

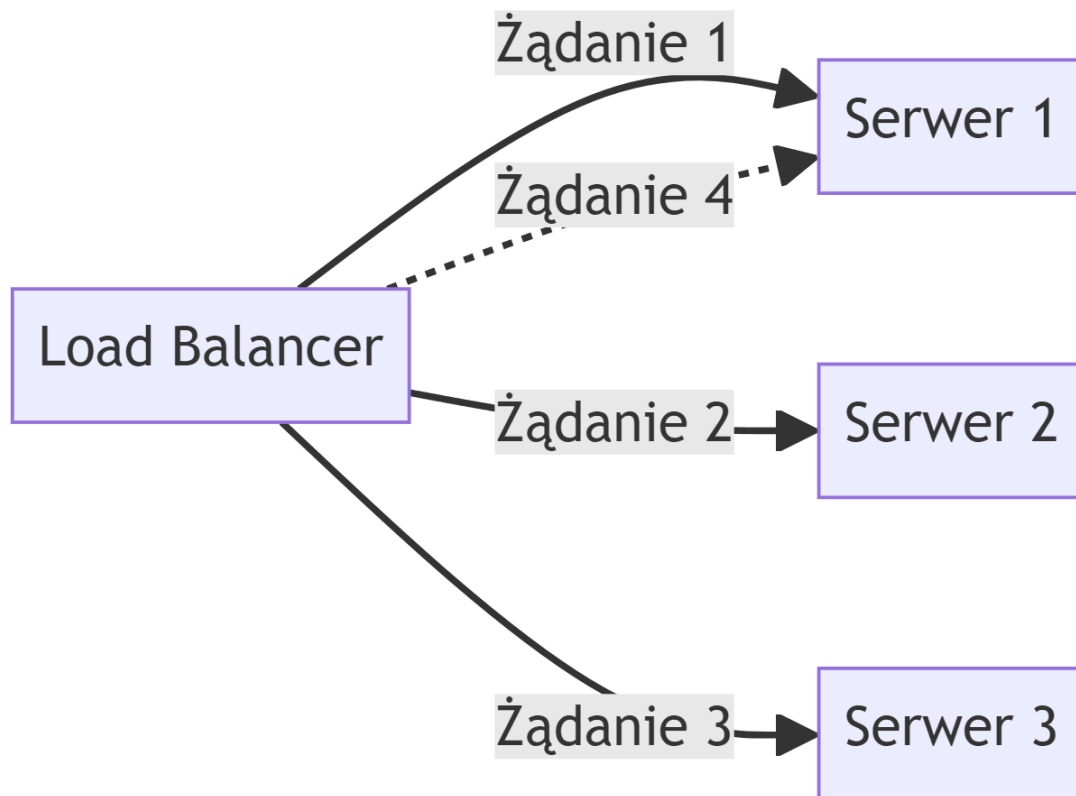
Instrukcja laboratoryjna 5	Przetwarzanie / Systemy odporne na błędy
	Temat: Wprowadzenie do systemów równomiernego obciążenia
	Przygotował: mgr inż. Michał Młodawski

1.Przykład zastosowania równomiernego obciążenia jako elementu systemu odpornego na błędy

Load balancer, czyli mechanizm równoważenia obciążenia, pełni przeważnie rolę centralnego punktu dystrybucji ruchu. Jego głównym zadaniem jest efektywne rozdzielanie przychodzących żądań między dostępne serwery, co przyczynia się do optymalizacji wykorzystania zasobów i poprawy wydajności całego systemu. Load balancer implementuje różne strategie dystrybucji ruchu, z których najpopularniejsze to:

1.1.Round Robin

To najprostsza metoda równoważenia obciążenia, polegająca na sekwencyjnym przydzielaniu żądań do kolejnych serwerów. Load balancer utrzymuje listę dostępnych serwerów i cyklicznie przydziela im nadchodzące żądania. Po dotarciu do końca listy, proces zaczyna się od początku. Metoda ta zapewnia równomierne rozłożenie obciążenia, zakładając, że wszystkie serwery mają podobną wydajność i że żądania wymagają podobnych zasobów do przetworzenia. Jest podstawowym narzędziem w dziedzinie równoważenia obciążenia, oferującym prostotę i niskie koszty implementacji. Jest szczególnie efektywna w środowiskach z homogenicznymi serwerami i jednolitym ruchem sieciowym. Jednak jej ograniczenia, takie jak brak uwzględnienia aktualnego obciążenia serwerów czy brak mechanizmów failover, sprawiają, że w bardziej złożonych i dynamicznych środowiskach konieczne jest zastosowanie bardziej zaawansowanych metod lub wariantów, takich jak Weighted Round Robin czy Dynamic Round Robin. W miarę rozwoju technologii i wzrostu wymagań użytkowników, strategia równoważenia obciążenia powinna być regularnie oceniana i dostosowywana do aktualnych potrzeb.



Zasada działania

- **Sekwencyjność:** Żądania są przydzielane do serwerów w ustalonej kolejności, bez względu na aktualne obciążenie czy wydajność poszczególnych serwerów.
- **Cykliczność:** Po obsłużeniu ostatniego serwera na liście, load balancer wraca do pierwszego, zapewniając ciągłość procesu.
- **Brak stanu:** Metoda nie wymaga przechowywania informacji o wcześniejszych przydziałach czy stanie serwerów, co upraszcza implementację.

Zalety metody Round Robin

1. **Prostota implementacji:** Ze względu na swoją prostotę, Round Robin jest łatwy do zaimplementowania w różnych środowiskach i nie wymaga skomplikowanych algorytmów ani dużych zasobów obliczeniowych.
2. **Równomierne rozłożenie obciążenia:** W środowiskach z homogenicznymi serwerami i jednolitym ruchem sieciowym metoda ta zapewnia równomierne rozłożenie obciążenia.
3. **Niskie koszty utrzymania:** Brak konieczności monitorowania stanu serwerów czy analizowania charakterystyki żądań obniża koszty operacyjne i upraszcza zarządzanie.

Wady metody Round Robin

1. **Brak uwzględnienia aktualnego obciążenia:** Metoda nie bierze pod uwagę rzeczywistego obciążenia serwerów ani ich wydajności, co może prowadzić do

nierównomiernego rozłożenia obciążenia w środowiskach z heterogenicznymi serwerami.

2. **Nieodpowiednia dla zróżnicowanych żądań:** Jeśli żądania różnią się znacznie pod względem wymagań zasobowych, Round Robin może nie zapewnić efektywnego równoważenia obciążenia.
3. **Brak mechanizmów failover:** Standardowa implementacja nie uwzględnia awarii serwerów. Jeśli serwer przestanie działać, load balancer nadal będzie do niego kierował żądania, co może skutkować utratą danych lub błędami w obsłudze.
4. **Problemy z utrzymaniem stanu sesji:** W aplikacjach wymagających utrzymania stanu sesji (np. sklepy internetowe), brak spójności w przydzielaniu żądań do tego samego serwera może prowadzić do problemów z obsługą użytkowników.

Warianty metody Round Robin

1. **Weighted Round Robin (WRR):** W tym wariantcie serwerom przypisywane są wagi odzwierciedlające ich wydajność lub zdolność obsługi ruchu. Load balancer przydziela więcej żądań serwerom o wyższej wadze, co pozwala lepiej wykorzystać zasoby w środowiskach z niejednorodnymi serwerami.
2. **Dynamic Round Robin:** Uwzględnia aktualne obciążenie serwerów, dynamicznie dostosowując wagi na podstawie bieżących statystyk. Jest to bardziej zaawansowana metoda, która wymaga monitorowania wydajności serwerów w czasie rzeczywistym.

Zastosowania metody Round Robin

- **Serwery WWW:** W środowiskach z identycznymi serwerami aplikacyjnymi, gdzie żądania są jednorodne.
- **Systemy DNS:** Round Robin jest często stosowany w DNS do rozkładania ruchu na różne adresy IP przypisane do jednej nazwy domenowej (DNS Round Robin).
- **Usługi streamingowe:** W dystrybucji treści multimedialnych, gdzie serwery dostarczają te same zasoby.

Techniczne aspekty implementacji

- **Sprzętowe load balancery:** Urządzenia takie jak F5 Big-IP czy Citrix ADC oferują wsparcie dla metody Round Robin na poziomie sprzętowym.
- **Oprogramowanie load balancery:** Narzędzia open-source, takie jak HAProxy, Nginx czy Apache mod_proxy_balancer, umożliwiają implementację Round Robin w konfiguracjach serwerowych.
- **DNS Round Robin:** Konfiguracja wielu rekordów A dla jednej nazwy domenowej, co pozwala na rotacyjne zwracanie różnych adresów IP w odpowiedziach DNS.

Problemy i wyzwania

- **Caching w DNS:** W przypadku DNS Round Robin, serwery cache'ujące mogą przechowywać adresy IP przez dłuższy czas, co zaburza równomierne rozłożenie ruchu.
- **Brak detekcji awarii:** W standardowym Round Robin brak jest mechanizmów sprawdzających dostępność serwerów, co może prowadzić do kierowania ruchu do niedostępnych maszyn.
- **Sesje użytkowników:** W aplikacjach stanowych konieczne jest zastosowanie dodatkowych mechanizmów, takich jak sticky sessions lub centralne przechowywanie stanu użytkownika.

Najlepsze praktyki

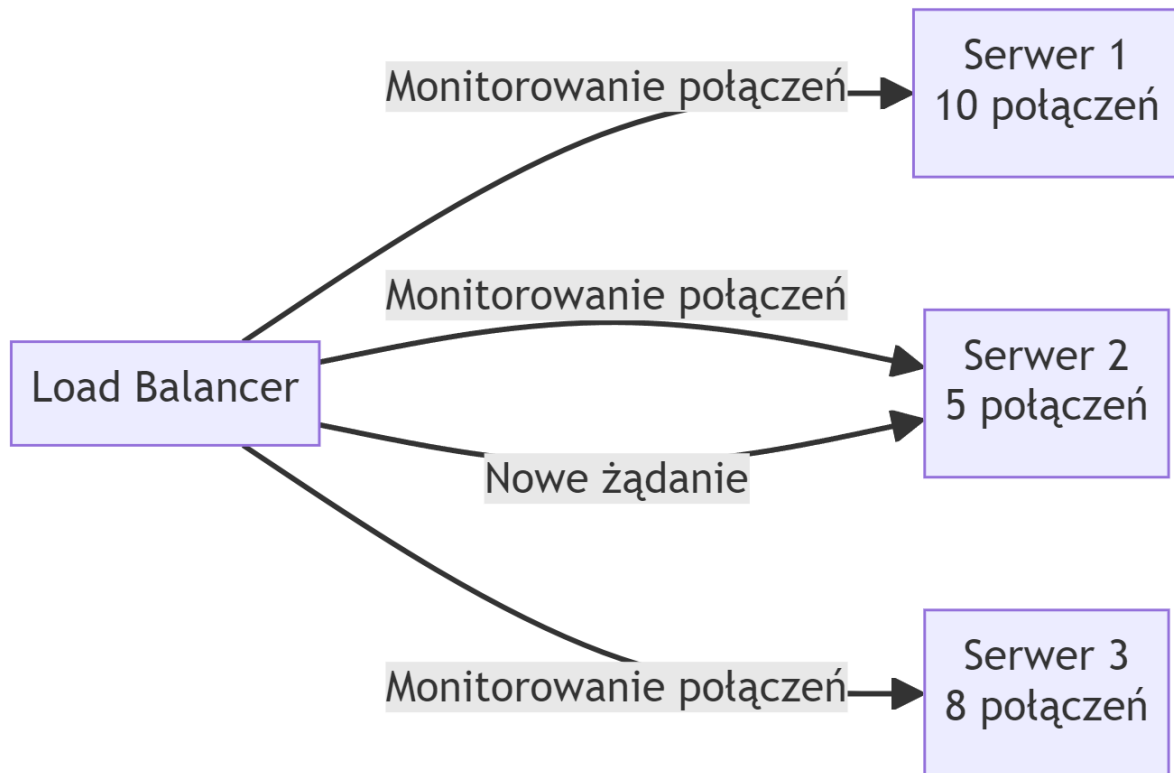
- **Monitorowanie zdrowia serwerów (health checks):** Implementacja regularnych testów dostępności serwerów i dynamiczne modyfikowanie listy dostępnych zasobów.
- **Skalowalność:** W miarę wzrostu ruchu i liczby serwerów, rozważenie przejścia na bardziej zaawansowane metody równoważenia obciążenia.
- **Konfiguracja wag:** W środowiskach z serwerami o różnej wydajności zastosowanie Weighted Round Robin.

Przykłady zastosowania

- **Usługi chmurowe:** W platformach takich jak Amazon Web Services czy Microsoft Azure, Round Robin może być używany w prostych konfiguracjach Load Balancerów.
- **Systemy CDN:** Sieci dostarczania treści mogą wykorzystywać Round Robin w połączeniu z innymi metodami do efektywnego rozkładania ruchu.

1.2.Least Connections

Jest to metoda, która kieruje nowe żądania do serwera obsługującego aktualnie najmniejszą liczbę aktywnych połączeń. Load balancer monitoruje liczbę otwartych połączeń na każdym serwerze i wybiera ten z najmniejszym obciążeniem dla każdego nowego żądania. Ta strategia jest szczególnie efektywna w sytuacjach, gdy czas przetwarzania żądań jest zróżnicowany lub nieprzewidywalny.



Metoda Least Connections działa na zasadzie dynamicznego przydzielania żądań, co pozwala na bardziej efektywne wykorzystanie zasobów serwerowych. W przeciwieństwie do statycznych metod, takich jak Round Robin, które nie uwzględniają aktualnego stanu obciążenia serwerów, Least Connections dostosowuje się do bieżących warunków w systemie. Dzięki temu minimalizuje ryzyko przeciążenia pojedynczych serwerów i zapewnia płynne działanie aplikacji. Jest to efektywna metoda do równoważenia obciążenia w dynamicznych i złożonych środowiskach. Dzięki uwzględnieniu aktualnego stanu serwerów pozwala na optymalne wykorzystanie zasobów i zapewnienie wysokiej dostępności usług. Mimo pewnych wyzwań związanych z jej implementacją, korzyści płynące z jej zastosowania czynią ją popularnym wyborem wśród administratorów sieci i architektów systemów.

Zalety metody Least Connections

Jedną z głównych zalet tej metody jest jej zdolność do równomiernego rozkładania obciążenia w środowiskach o zmiennym i nieprzewidywalnym ruchu sieciowym. Ponieważ uwzględnia aktualną liczbę aktywnych połączeń, może skutecznie reagować na nagłe wzrosty obciążenia i dystrybuować ruch w sposób optymalny. Jest to szczególnie ważne w aplikacjach, gdzie niektóre żądania mogą wymagać znacznie więcej zasobów niż inne.

Wady metody Least Connections

Mimo wielu zalet, metoda ta ma też pewne ograniczenia. Wymaga ciągłego monitorowania liczby aktywnych połączeń na każdym serwerze, co może zwiększyć obciążenie samego load balancera i skomplikować jego implementację. Ponadto, w środowiskach z bardzo krótkotrwałymi połączeniami, różnice w liczbie aktywnych połączeń między serwerami mogą być minimalne, co ogranicza efektywność tej metody.

Warianty metody Least Connections

Aby zwiększyć efektywność i dostosować metodę do różnych środowisk, opracowano jej różne warianty:

- **Weighted Least Connections:** W tym podejściu serwerom przypisywane są wagi odzwierciedlające ich wydajność lub zdolności sprzętowe. Load balancer bierze pod uwagę zarówno liczbę aktywnych połączeń, jak i wagę serwera, co pozwala lepiej wykorzystać mocniejsze maszyny.
- **Adaptive Least Connections:** Ten wariant uwzględnia dodatkowe metryki, takie jak średni czas odpowiedzi serwera, co pozwala na jeszcze bardziej precyzyjne równoważenie obciążenia.

Zastosowania metody Least Connections

Metoda Least Connections jest szeroko stosowana w środowiskach, gdzie obciążenie jest nierównomierne i trudne do przewidzenia. Przykłady obejmują:

- **Serwery aplikacji webowych:** Gdzie żądania mogą znacznie różnić się pod względem złożoności i czasu przetwarzania.
- **Usługi streamingowe:** Gdzie liczba aktywnych połączeń może dynamicznie się zmieniać.
- **Systemy e-commerce:** Gdzie ruch może być niestabilny, zwłaszcza podczas promocji czy sezonowych wyprzedaży.

Techniczne aspekty implementacji

Implementacja metody Least Connections wymaga:

- **Ciągłego monitorowania:** Load balancer musi stale aktualizować informacje o liczbie aktywnych połączeń na każdym serwerze.
- **Skalowalności:** Load balancer sam musi być wydajny, aby nie stać się wąskim gardłem systemu.
- **Bezpieczeństwa i niezawodności:** W przypadku awarii serwera, load balancer musi szybko przekierować ruch do innych dostępnych serwerów.

Problemy i wyzwania

- **Synchronizacja danych:** W systemach z wieloma load balancerami konieczne jest synchronizowanie informacji o połączeniach między nimi.
- **Obciążenie load balancera:** Dodatkowe obliczenia mogą wpłynąć na jego wydajność.
- **Krótkotrwałe połączenia:** W systemach z bardzo krótkimi połączeniami metoda może nie przynosić oczekiwanych korzyści.

Porównanie z innymi metodami

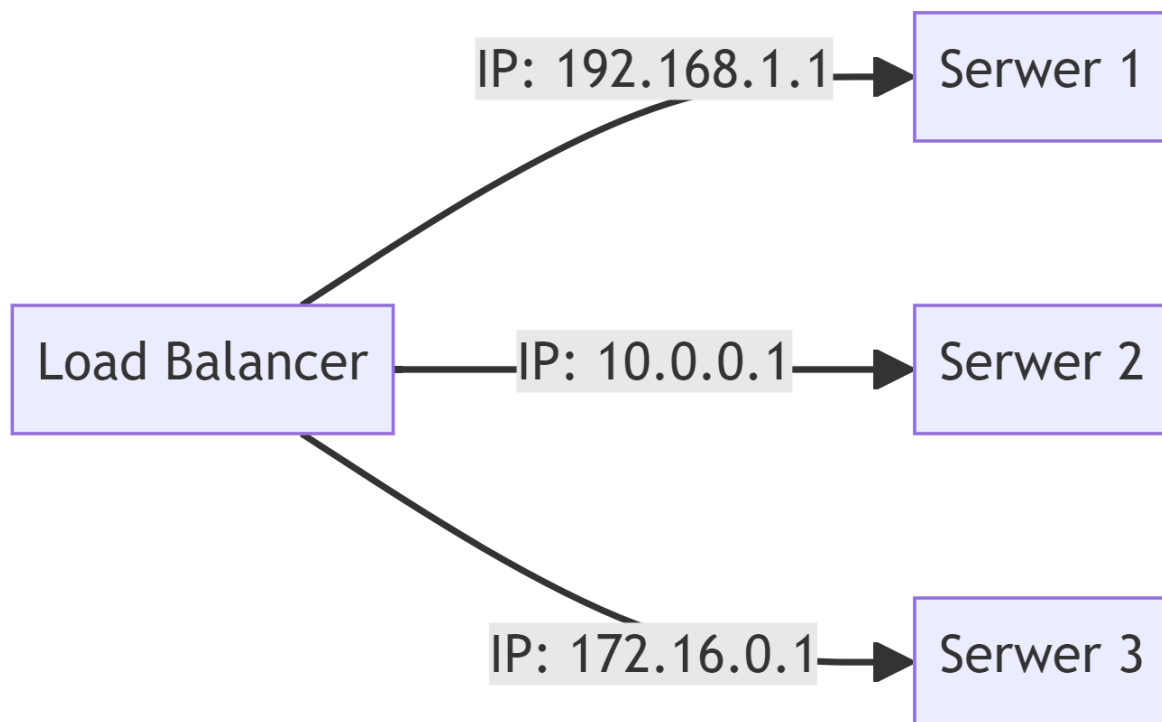
W porównaniu z metodą Round Robin, Least Connections oferuje bardziej dynamiczne i adaptacyjne podejście. Jednak w porównaniu z metodami opartymi na bardziej zaawansowanych algorytmach, takimi jak **Least Response Time** czy **Hashing**, może być mniej skuteczna w pewnych specyficznych scenariuszach.

Najlepsze praktyki

- **Monitorowanie wydajności:** Regularne sprawdzanie działania load balancera i serwerów backendowych.
- **Optymalizacja konfiguracji:** Dostosowanie parametrów load balancera do specyfiki ruchu i aplikacji.
- **Redundancja:** Zapewnienie dodatkowych zasobów na wypadek awarii.

1.3.IP Hash

Jest to metoda, która przypisuje żądania do serwerów na podstawie adresu IP klienta. Load balancer wykonuje operację haszowania na adresie IP źródłowym żądania i na podstawie wyniku przydziela je do konkretnego serwera. Ta metoda zapewnia spójność sesji, co oznacza, że kolejne żądania od tego samego klienta będą zawsze kierowane do tego samego serwera.



Zastosowanie adresu IP jako klucza do haszowania pozwala na deterministyczne rozdzielanie żądań między serwery. Jest to szczególnie przydatne w aplikacjach, gdzie ważne jest utrzymanie stanu sesji po stronie serwera, takich jak sklepy internetowe czy usługi bankowe. Użycie tej metody jest skutecznym rozwiązaniem w sytuacjach, gdzie kluczowe jest utrzymanie spójności sesji bez konieczności dodatkowego przechowywania stanu po stronie load balancera. Jednak jej efektywność może być ograniczona w środowiskach z dużą liczbą klientów za NAT-em lub przy dynamicznie zmieniającej się infrastrukturze serwerowej. Dlatego ważne jest świadome zastosowanie tej metody, uwzględniając specyfikę środowiska oraz potencjalne wyzwania, i ewentualne łączenie jej z innymi technikami równoważenia obciążenia.

Zasada działania

Gdy klient wysyła żądanie, load balancer pobiera adres IP źródłowy i stosuje na nim funkcję haszującą. Wynik tej funkcji jest następnie mapowany na jeden z serwerów w puli. Dzięki temu ten sam adres IP klienta zawsze trafia na ten sam serwer, co umożliwia utrzymanie stanu sesji bez konieczności dodatkowego przechowywania informacji o sesji w innych miejscach.

Jeśli jeden z serwerów stanie się niedostępny, mapowanie zostanie zmienione, co może skutkować przekierowaniem klientów na inne serwery i potencjalną utratą stanu sesji.

Zalety metody IP Hash

1. **Spójność sesji:** Zapewnia, że żądania od tego samego klienta trafiają zawsze do tego samego serwera, co jest kluczowe w aplikacjach stanowych.
2. **Prostota implementacji:** Nie wymaga przechowywania dodatkowych informacji o stanie sesji czy klientach, co upraszcza konfigurację load balancera.
3. **Deterministyczność:** Metoda jest przewidywalna i nie zależy od zmiennych czynników, takich jak obciążenie serwerów czy liczba aktywnych połączeń.

Wady metody IP Hash

1. **Nierównomierne rozłożenie obciążenia:** Jeśli duża liczba klientów pochodzi z tego samego adresu IP (np. za NAT-em), może to prowadzić do przeciążenia jednego serwera.
2. **Brak elastyczności:** Nie uwzględnia aktualnego obciążenia czy wydajności serwerów, co może skutkować nieoptymalnym wykorzystaniem zasobów.
3. **Problem przy zmianie puli serwerów:** Dodanie lub usunięcie serwera zmienia mapowanie adresów IP na serwery, co może prowadzić do utraty stanu sesji dla części klientów.

Zastosowania metody IP Hash

- **Aplikacje stanowe:** Gdzie istotne jest utrzymanie stanu sesji po stronie serwera bez dodatkowych mechanizmów.
- **Usługi finansowe i e-commerce:** Gdzie ciągłość sesji wpływa na bezpieczeństwo i wygodę użytkowników.
- **Systemy z unikalnymi adresami IP klientów:** Gdzie ryzyko nierównomiernego rozłożenia obciążenia jest mniejsze.

Techniczne aspekty implementacji

- **Wybór funkcji haszującej:** Musi zapewniać równomierne rozłożenie adresów IP na dostępne serwery.
- **Obsługa IPv6:** Funkcja haszująca powinna być kompatybilna zarówno z IPv4, jak i IPv6.

- **Mechanizmy failover:** Konieczne jest zapewnienie, że w przypadku awarii serwera klienci zostaną przekierowani na inny dostępny serwer.

Problemy i wyzwania

- **Klienci za NAT-em:** Wielu użytkowników może dzielić ten sam adres IP, co prowadzi do przeciążenia jednego serwera.
- **Zmiany w infrastrukturze:** Dodanie lub usunięcie serwerów może zmienić wyniki funkcji haszującej, wpływając na spójność sesji.
- **Brak uwzględnienia obciążenia:** Metoda nie reaguje na aktualne obciążenie serwerów, co może skutkować nierównomiernym rozłożeniem ruchu.

Porównanie z innymi metodami równoważenia obciążenia

- **Round Robin:** Przydziela żądania cyklicznie do kolejnych serwerów, nie zapewniając spójności sesji.
- **Least Connections:** Kieruje żądania do serwera z najmniejszą liczbą aktywnych połączeń, lepiej rozkładając obciążenie, ale bez gwarancji spójności sesji.
- **Weighted Round Robin:** Uwzględnia wagi serwerów, ale również nie zapewnia przypisania klienta do stałego serwera.

Najlepsze praktyki

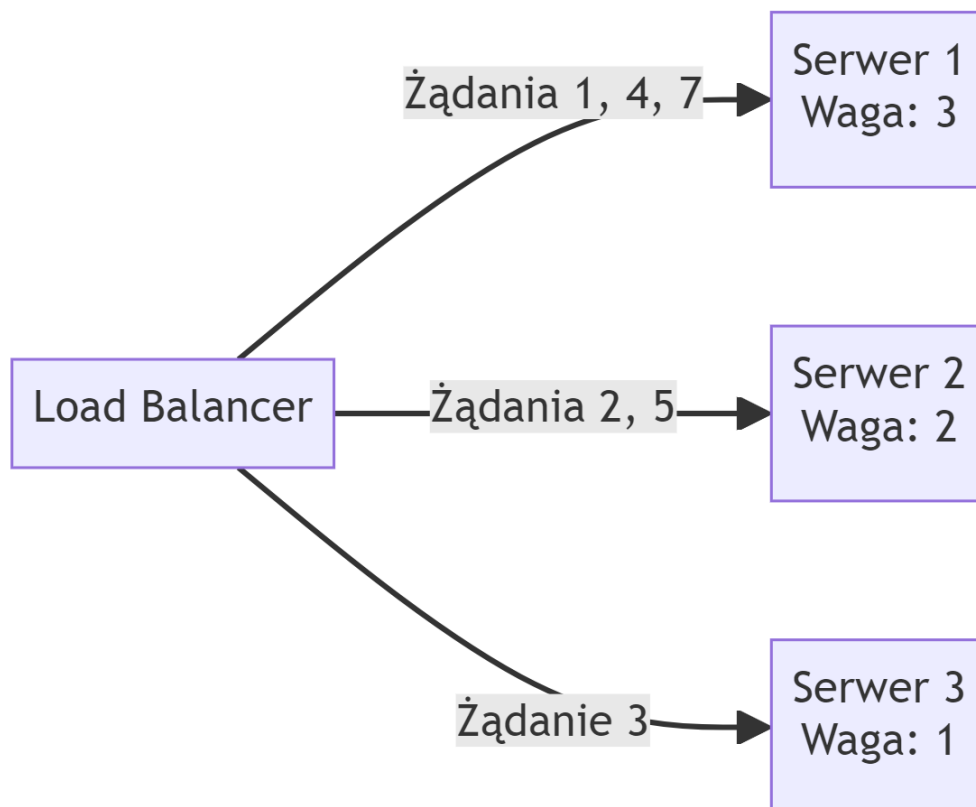
- **Wykorzystanie haszowania konsystentnego (consistent hashing):** Minimalizuje wpływ zmian w puli serwerów na mapowanie adresów IP.
- **Implementacja mechanizmów redundancji sesji:** Przechowywanie stanu sesji w zewnętrznych magazynach, takich jak bazy danych czy systemy cache (np. Redis).
- **Monitorowanie obciążenia:** Regularne sprawdzanie rozkładu ruchu i dostosowywanie konfiguracji w celu uniknięcia przeciążenia.

Przyszłe kierunki rozwoju

- **Integracja z technologiami chmurowymi:** Automatyczne skalowanie i dostosowywanie puli serwerów bez utraty spójności sesji.
- **Zaawansowane algorytmy haszujące:** Wykorzystanie bardziej skomplikowanych funkcji haszujących, które lepiej rozkładają ruch i są odporne na zmiany w infrastrukturze.
- **Łączenie metod:** Kombinacja IP Hash z innymi metodami równoważenia obciążenia w celu optymalizacji zarówno spójności sesji, jak i rozkładu obciążenia.

1.4.Weighted Round Robin

Weighted Round Robin to rozszerzenie metody Round Robin, które uwzględnia różnice w wydajności serwerów poprzez przypisanie im wag. Serwery o większej mocy obliczeniowej otrzymują wyższe wagi, co oznacza, że będą obsługiwać proporcjonalnie więcej żądań. Load balancer przydziela żądania do serwerów w sposób cykliczny, ale liczba żądań kierowanych do danego serwera jest proporcjonalna do jego wagi.



Weighted Round Robin to rozszerzenie klasycznej metody Round Robin, które uwzględnia różnice w wydajności serwerów poprzez przypisanie im odpowiednich wag. W tej metodzie serwery o większej mocy obliczeniowej otrzymują wyższe wagi, co oznacza, że będą obsługiwać proporcjonalnie więcej żądań. Load balancer przydziela żądania do serwerów w sposób cykliczny, ale liczba żądań kierowanych do danego serwera jest proporcjonalna do jego wagi. Weighted Round Robin to efektywna metoda równoważenia obciążenia w środowiskach z serwerami o różnej wydajności. Dzięki przypisaniu wag, umożliwia lepsze wykorzystanie zasobów i zwiększa ogólną wydajność systemu. Jednak aby metoda ta była skuteczna, wymaga regularnego monitorowania i aktualizacji wag, co może stanowić wyzwanie w dynamicznych środowiskach. Pomimo tych wyzwań, Weighted Round Robin pozostaje popularnym wyborem wśród administratorów sieci ze względu na swoją prostotę i skuteczność w wielu scenariuszach.

Metoda ta działa na zasadzie przypisania każdemu serwerowi liczbowej wagi, która odzwierciedla jego zdolność do obsługi ruchu. Na przykład, jeśli mamy trzy serwery o wagach

5, 3 i 2, to serwer z wagą 5 będzie otrzymywał 50% wszystkich żądań, serwer z wagą 3 otrzyma 30%, a serwer z wagą 2 pozostałe 20%. Load balancer utrzymuje listę serwerów wraz z ich wagami i dystrybuuje żądania zgodnie z tymi proporcjami.

Zalety metody Weighted Round Robin

Jedną z głównych zalet tej metody jest efektywne wykorzystanie zasobów w środowiskach z serwerami o różnej wydajności. Dzięki przypisaniu wag, mocniejsze serwery mogą obsługiwać większą część ruchu, co zwiększa ogólną wydajność systemu. Metoda ta zachowuje prostotę i przewidywalność klasycznego Round Robin, jednocześnie dodając możliwość lepszego dostosowania do specyfiki infrastruktury.

Wady metody Weighted Round Robin

Mimo swoich zalet, metoda Weighted Round Robin ma też pewne ograniczenia. Nie uwzględnia ona dynamicznych zmian w obciążeniu serwerów ani ich aktualnej wydajności. Jeśli wydajność serwera ulegnie zmianie z powodu awarii sprzętowej czy zwiększonego obciążenia, metoda nie dostosuje automatycznie przypisanych wag. Wymaga to ręcznej interwencji administratora lub zastosowania dodatkowych mechanizmów monitorujących i aktualizujących wagi w czasie rzeczywistym.

Zastosowania metody Weighted Round Robin

Weighted Round Robin znajduje zastosowanie w wielu środowiskach, gdzie serwery różnią się pod względem wydajności lub zasobów. Przykłady obejmują:

- **Centra danych z heterogenicznym sprzętem:** Gdzie różne serwery mają różne specyfikacje techniczne i możliwości obliczeniowe.
- **Migracje systemów:** Kiedy nowe, bardziej wydajne serwery są stopniowo wprowadzane do istniejącej infrastruktury.
- **Usługi chmurowe:** Gdzie zasoby mogą być dynamicznie alokowane, a wagi serwerów dostosowywane do aktualnej dostępności zasobów.

Techniczne aspekty implementacji

Implementacja Weighted Round Robin wymaga:

- **Przypisania wag serwerom:** W oparciu o ich wydajność, takie jak liczba rdzeni CPU, ilość pamięci RAM czy przepustowość sieci.
- **Konfiguracji load balancera:** Który musi obsługiwać tę metodę i umożliwiać ustawienie wag dla każdego serwera.
- **Monitorowania i aktualizacji wag:** Aby zapewnić, że wagi odzwierciedlają aktualny stan i wydajność serwerów.

Problemy i wyzwania

- **Statyczne wagi:** Jeśli wagi nie są regularnie aktualizowane, mogą nie odzwierciedlać rzeczywistej wydajności serwerów, prowadząc do nierównomiernego obciążenia.
- **Brak reakcji na dynamiczne obciążenie:** Metoda nie uwzględnia bieżącego obciążenia serwerów, co może być problematyczne w środowiskach z szybko zmieniającym się ruchem.
- **Skalowalność:** W dużych infrastrukturach zarządzanie wagami może stać się skomplikowane i wymagać dodatkowych narzędzi do automatyzacji.

Porównanie z innymi metodami równoważenia obciążenia

- **Round Robin:** Prosta metoda przydzielająca żądania równomiernie, nie uwzględniając różnic w wydajności serwerów.
- **Least Connections:** Kieruje żądania do serwera z najmniejszą liczbą aktywnych połączeń, uwzględniając bieżące obciążenie.
- **Weighted Least Connections:** Kombinuje wagi serwerów z liczbą aktywnych połączeń, zapewniając bardziej dynamiczne rozłożenie obciążenia.

Najlepsze praktyki

- **Regularne monitorowanie serwerów:** Aby wagi odzwierciedlały aktualną wydajność i dostępność.
- **Automatyzacja aktualizacji wag:** W miarę możliwości zastosowanie narzędzi automatyzujących proces dostosowywania wag.
- **Integracja z systemami monitoringu:** Takimi jak Nagios czy Zabbix, w celu szybkiego wykrywania zmian w wydajności serwerów.

Każda z tych metod ma swoje zastosowania i ograniczenia. Round Robin jest prosta i efektywna dla jednolitych systemów. Least Connections sprawdza się w środowiskach o zmiennym obciążeniu. IP Hash jest użyteczna dla aplikacji wymagających spójności sesji. Weighted Round Robin pozwala na optymalne wykorzystanie heterogenicznych zasobów. Wybór odpowiedniej metody zależy od specyfiki systemu, charakterystyki ruchu oraz wymagań aplikacji. W praktyce, zaawansowane systemy równoważenia obciążenia często łączą różne metody lub wykorzystują bardziej złożone algorytmy, adaptujące się do zmieniających się warunków w czasie rzeczywistym.

2.Przykładowa implementacja load balancer w Pythonie.

W przykładowym rozwiązaniu zastosowano wariant algorytmu Load balancer oparty na identyfikatorze sesji, co pozwala na zachowanie spójności sesji użytkownika przy jednoczesnym rozproszeniu obciążenia.

Istotnym elementem systemu jest mechanizm health check, który regularnie sprawdza stan serwerów. Dzięki temu load balancer może dynamicznie dostosowywać listę aktywnych serwerów, eliminując z puli te, które przestały odpowiadać lub działają nieprawidłowo. To podejście znacząco zwiększa niezawodność systemu, automatycznie izolując problematyczne komponenty.

Przedstawione rozwiązanie implementuje również koncepcję odporności na przeciążenia poprzez mechanizm rate limiting. Ograniczając liczbę żądań od pojedynczego klienta w określonym czasie, system chroni się przed potencjalnymi atakami typu DoS (Denial of Service) oraz zapewnia sprawiedliwy dostęp do zasobów dla wszystkich użytkowników.

Przykładowy program jest to rozbudowany system odporny na awarie, zaimplementowany w Pythonie, który wykorzystuje load balancer, wiele serwerów i graficzny interfejs użytkownika (GUI) do kontroli i wizualizacji.

