Wrocław University of Science and Technology

Przetwarzanie danych masowych

Wykład 8 – Produkcyjne aspekty utrzymywania i wdrażania aplikacji

dr inż. Tomasz Kajdanowicz, Roman Bartusiak, Piotr Bielak, Krzysztof Rajda

6 grudnia 2021 r.

HR EXCELLENCE IN RESEARCH

Wrocław University of Science and Technology

Spis treści

Wrocław University of Science and Technology

Spis treści

Wprowadzenie

Aplikacje stanowe i bezstanowe

- Jak projektować aplikacje?
- Jak zapisywać dane o sesjach użytkowników?
- Jak zapisywać dane o użytkownikach?

Bezstanowość (1)

Aplikacje stanowe i bezstanowe

- Aplikacje nie przechowują w pamięci żadnych (dodatkowych) danych
- Zapytania są przetwarzane pojedynczo
- Zapytania są przetwarzane niezależnie, bez uwzględnianie kontekstu innych zapytań
- Sesje nie są zapisywane w pamięci aplikacji

Bezstanowość (2)

Aplikacje stanowe i bezstanowe

- Sesje mogą być zapisywane w zewnętrznym serwisie (np. bazie danych)
  - Dostępny dla innych kopii aplikacji (ang. *replicas*)
  - np. Redis, MongoDB
  - Może być również skalowana (niezależnie od aplikacji)
- Brak potrzeby tzw. mechanizmu *sticky session*
- Aplikacje są łatwo skalowalne horyzontalnie
- Implementacja aplikacji może być bardziej skomplikowana

Wrocław University of Science and Technology

Stanowość (1)

Aplikacje stanowe i bezstanowe

- Aplikacje mogą zapisywać (dodatkowe) dane w pamięci
- Zapytania mogą być przetwarzane w kontekście poprzednich zapytań
- Sesja jest zapisywana w pamięci aplikacji

Stanowość (2)

Aplikacje stanowe i bezstanowe

- Potrzeba użycia mechnizmu *sticky session*
- Skalowanie i dostępność aplikacji nie są trywialne
- Implementacja aplikacji może być łatwiejsza

Wrocław University of Science and Technology

Spis treści

Wprowadzenie

Wysoka dostępność (ang. *High Availability*)

- cecha aplikacji wdrożonych w środowiskach produkcyjnych
- aplikacja powinna być jak najdłużej dostępna
- SLA (Service Level Agreement) - *"dziewiątki"*

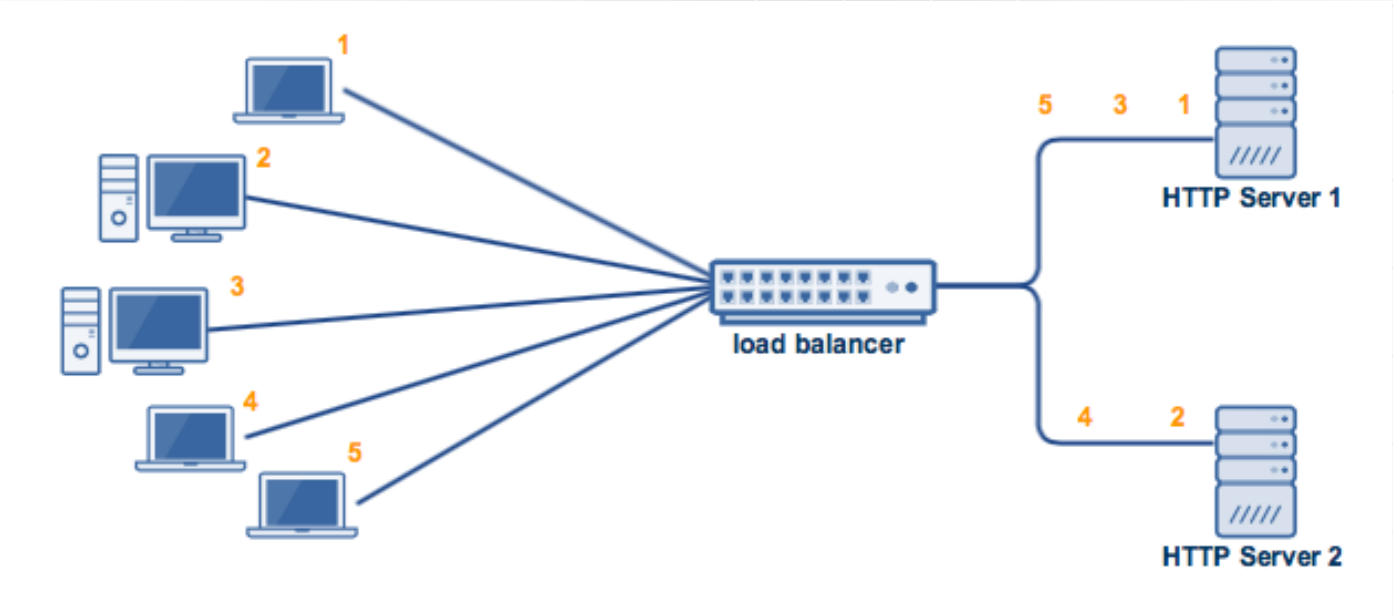
Availability %	Downtime per year[note 1]	Downtime per month	Downtime per week	Downtime per day
55.5555555% ("nine fives")	162.33 days	13.53 days	74.92 hours	10.67 hours
90% ("one nine")	**36.53 days**	**73.05 hours**	**16.80 hours**	**2.40 hours**
95% ("one nine five")	18.26 days	36.53 hours	8.40 hours	1.20 hours
97%	10.96 days	21.92 hours	5.04 hours	43.20 minutes
98%	7.31 days	14.61 hours	3.36 hours	28.80 minutes
99% ("two nines")	**3.65 days**	**7.31 hours**	**1.68 hours**	**14.40 minutes**
99.5% ("two nines five")	1.83 days	3.65 hours	50.40 minutes	7.20 minutes
99.8%	17.53 hours	87.66 minutes	20.16 minutes	2.88 minutes
99.9% ("three nines")	**8.77 hours**	**43.83 minutes**	**10.08 minutes**	**1.44 minutes**
99.95% ("three nines five")	4.38 hours	21.92 minutes	5.04 minutes	43.20 seconds
99.99% ("four nines")	**52.60 minutes**	**4.38 minutes**	**1.01 minutes**	**8.64 seconds**
99.995% ("four nines five")	26.30 minutes	2.19 minutes	30.24 seconds	4.32 seconds
99.999% ("five nines")	**5.26 minutes**	**26.30 seconds**	**6.05 seconds**	**864.00 milliseconds**
99.9999% ("six nines")	**31.56 seconds**	**2.63 seconds**	**604.80 milliseconds**	**86.40 milliseconds**
99.99999% ("seven nines")	**3.16 seconds**	**262.98 milliseconds**	**60.48 milliseconds**	**8.64 milliseconds**
99.999999% ("eight nines")	**315.58 milliseconds**	**26.30 milliseconds**	**6.05 milliseconds**	**864.00 microseconds**
99.9999999% ("nine nines")	**31.56 milliseconds**	**2.63 milliseconds**	**604.80 microseconds**	**86.40 microseconds**

Zarys HA

Wysoka dostępność (ang. *High Availability*)

Jak uzyskać HA?

- zreplikuj serwis
- skieruj cały ruch przez proxy / load balancer



HA w rzeczywistych systemach

Wysoka dostępność (ang. *High Availability*)

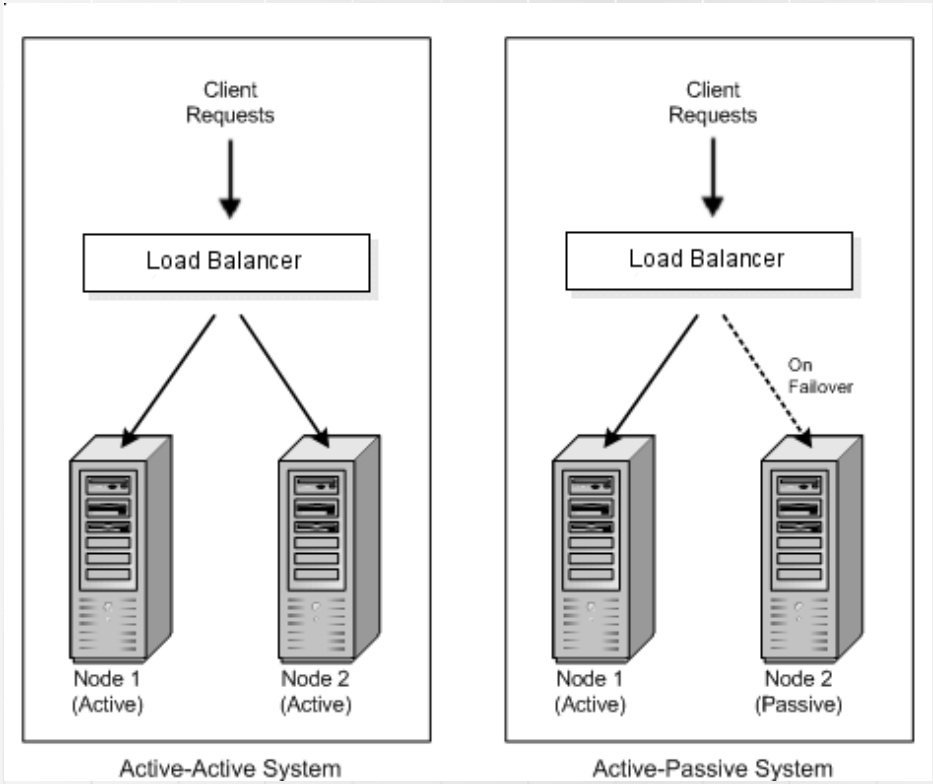
W rzeczywistości nie jest to takie proste:

- implementacja próbek dostępności (ang. *health checks*, *liveness probes*),
- mechanizmy ponawiania (ang. *retry*),
- jak automatycznie skalować serwisy?
- wybór kryterium skalowania, liczby replik awaryjnych (ang. *fallback replicas*),
- zapewnienie szybkiego uruchamiania aplikacji,

Active-active vs active-passive

Wysoka dostępność (ang. *High Availability*)

- dwa typy HA: **active-active**, **active-passive**
- dla aplikacji stanowych:
  - active-active is trudny do utrzymywania $\rightarrow$ stan wszystkich instancji musi być synchronizowany wraz z kolejnymi zapytaniami,
  - active-passive również wymaga synchronizacji stanu (prościej, bo tylko jeden węzeł przetwarza zapytania),
- aplikacje bezstanowe – oba typy HA są łatwiejsze w zastosowaniu,



Wrocław University of Science and Technology

Spis treści

Wprowadzenie

Continuous X

- Jak współpracować w ramach zespołu?
- Jak zapewnić, że aplikacje działają, a kod jest wysokiej jakości?
- Jak sprawdzić, że coś zostało nie działa / zostało zepsute?
- Co jeśli dwie osoby modyfikują zależne komponenty?

- dzisiaj o Continuous Integration, późniejszym wykładzie o Continuous Delivery

Continuous Integration (1)

Continuous X

Pojęcie wprowadzone w *Extreme Programming* autorstwa Kenta Becka.

- testowanie kodu jest ważne!
- rożne rodzaje testów:
  - jednostkowe,
  - integracyjne,
  - funkcjonalne,
  - regresyjne,
  - jakość kodu,
  - bezpieczeństwo

Continuous Integration (2)

Continuous X

- manualne a automatyczne testy,
- manualne:
  - nie są idealnym rozwiązaniem, ale wciąż używane
  - np. wykonywane przez testerów w celu przygotowania testów automatycznych
  - np. testy akceptacyjne po stronie klienta
- automatyczne:
  - używają bibliotek do testowania (np. w Pythonie: unittest, pytest, flake8, w Scali: ScalaTest, Play),
  - powinny utworzyć odpowiednie (reprodukowalne) środowiska testowe,
  - można je uruchomić na komputerach programistów,
  - po opublikowaniu kodu w repozytorium, system CI powinien zbudować kod i uruchomić wszystkie testy (ang. *CI pipeline*)

Serwery automatyzacji

Continuous X

- **Jenkins**
- **GitLab CI**
- TeamCity
- **Zuul**
- Travis
- CircleCI

Wrocław University of Science and Technology

Spis treści

Wprowadzenie

Aktualizacje aplikacji

- Jak wykonywać aktualizacje aplikacji bez przerwy w dostępności (tzw. **downtime**)?
- Jak wykonywać aktualizację dla wybranej części użytkowników?
- Jak zapewnić i sprawdzić czy aplikacja będzie działać poprawnie przed pełną aktualizacją?

Typy procesu aktualizacji

Aktualizacje aplikacji

- Blue-Green
- Canary
- Rolling

Aktualizacja schematów danych (1)

Aktualizacje aplikacji

- Jak obsługiwać warstwę zapisu danych (np. bazy danych)?
- Schematy danych muszą być odizolowane od aktualizacji aplikacji
- W przypadku zmian schematu możliwe jest, że proces aktualizacji będzie wykonywany w kilku etapach

Aktualizacja schematów danych (2)

Aktualizacje aplikacji

Usunięcie pola:

- Schemat i aplikacja wymagają pewnego pola
- Zmień schemat i wprowadź w aplikacji zmiany aby pole było opcjonalne
- Przeprowadź aktualizację
- Wyłącz instancje aplikacji, które wymagają tego pola
- Usuń pole ze schematu i kodu aplikacji

Blue-green

Aktualizacje aplikacji

- Przed aplikacją znajduje się proxy albo load-balancer
- Dwa takie same środowiska
  - Green - Obecnie używane środowisko
  - Blue - Środowisko, które jest aktualizowane
- Przekierowanie ruchu z *green* do *blue* po aktualizacji
- Jeśli wszystko działa: *blue* staje się *green*, natomiast **green** staje się **blue**
- Jeśli wykryto problemy: przywróć ruch do *green*

Canary (1)

Aktualizacje aplikacji

- Udostępnij zaktualizowaną aplikację tylko dla małego podzbioru ruchu / zapytań
- Można udostępnić dla konkretnych użytkowników / grupy użytkowników
- Facebook:
  - Gatekeeper
  - decyzja na podstawie informacji o użytkownikach
  - np. wiek, płeć

Canary (2)

Aktualizacje aplikacji

Procedura aktualizacji:

- Utwórz instancję z nową wersją aplikacji
- Przekieruj część ruchu (na podstawie wybranych kryteriów) do nowej instancji
- Wszystko w porządku: zwiększaj ilość przekierowywanego ruchu aż 100% ruchu będzie skierowane na nową wersję aplikacji
- Wystąpił problem: przekieruj ruch z powrotem na instancje ze starą wersją aplikacji

Rolling (1)

Aktualizacje aplikacji

- N - liczba instancji przed aktualizacją
- Podczas aktualizacji będzie uruchomionych co najwyżej $N + 1$ instancji
- W każdym kroku aktualizacji podmień jedną instancję ze starą wersją aplikacji na instancję z nową wersją aplikacji

Rolling (2)

Aktualizacje aplikacji

Procedura aktualizacji:

- Utwórz instancję w nową wersją aplikacji
- Wszystko w porządku: Wyłącz jedną instancję ze starą wersją aplikacji
- Wszystko w porządku: Powtarzaj aż wszystkie instancje będą zawierać nową wersję aplikacji
- Wystąpił problem: Uruchom ponownie instancje ze starą wersją aplikacji

Wrocław University of Science and Technology

Przetwarzanie danych masowych

Wykład 8 – Produkcyjne aspekty utrzymywania i wdrażania aplikacji

dr inż. Tomasz Kajdanowicz, Roman Bartusiak, Piotr Bielak, Krzysztof Rajda

6 grudnia 2021 r.