Sprawozdanie laboratoryjne
Nierelacyjne bazy danych

Nazwa projektu

Grupa:
1. Szymon, Szymajda, 210339
2. Zbigniew, Nowacki, 210284
3. Oleksandr, Dubenko, 211651
4. Yaroslav, Horetskyi, 211714

# Spis treści

# 1 Analiza problemu

## 1.1 Opis problemu do rozwiązania

Wymagania firmowe:

- firma nie posiada swojej infrastruktury i zamierza korzystać z rozwiązań chmurowych
- wymogiem jest aby baza danych mogła pracować w klastrze, tak aby zapewnić wysoką niezawodność oraz zachować dostępność do danych mimo wyłączenia/awarii jednego z węzłów klastra - odporność na błędy
- firma przewiduje znacznie większą liczbę zapisów do bazy danych niż odczytów. Zapisy są praktycznie w trybie ciągłym, analiza danych odbywa się w cyklu tygodniowym/miesięcznym
- firma zamierza rozwijać się i stać się firmą globalną. Aplikacja powinna działać "sprawnie", dlatego też należy przewidzieć możliwość umieszczenia jej i jej bazy danych w kilku centrach chmurowych, tak aby była ona jak najbliżej końcowego użytkownika
- przewidywany jest także duży wzrost transakcji bazodanowych, więc system baz danych musi przewidywać skalowanie rozwiązania w trakcie cyklu życia aplikacji

Informacje dodatkowe:

Można sobie wyobrazić, że firma planuje zbudować:

- system sprzedażowy i chce zapisywać wszystkie operacje wykonywane na towarach ze wszystkich swoich oddziałów na świecie
- system zakładów bookmacherskich, gdzie konieczne jest zbieranie i analiza danych/statystyk meczów z całego świata
- system zbierający i analizujący dane z urządzeń typu IoT, np. czujników monitorujących stan montowanych w pojazdach ciężarowych, osobowych, maszynach przemysłowych itd.
- zbierać informacje o warunkach pogodowych na świecie

Podczas analizy rozwiązania przyjmij jedną z powyższych propozycji aplikacji, bądź zaproponuj rozwiązanie o podobnym charakterze. Musi ono zawierać kluczowe wymagania firmowe

## 1.2 Opis dużego zbioru danych testowych

Źródło danych, opis struktur danych.

Wybrany zbiór danych testowych to cogodzinnie zbierane dane ze 122 stacji pogodowych w południowym regionie Brazylii w latach 2000-2016. Pochodzi z Narodowego Instytutu Meteorologicznego Brazylii (INMET).

Źródło: https://www.kaggle.com/PROPPG-PPG/hourly-weather-surface-brazil-southeast-region

## 1.3 Analiza i wybór systemów baz danych

Lista dostępnych systemów baz danych, które zostaną poddane analizie (minimum 3)

1. MongoDB
2. Cassandra
3. HBase
4. Redis

Analiza systemów baz danych pod kątem spełnienia wymagań: w pierwszej kolumnie wymagania co do aplikacji i systemu baz danych, w pierwszym wierszu analizowane systemy baz danych. Na przecięciach informację o tym czy dany system baz danych spełnia wymagania czy też nie, w jakim stopniu i czy przewidywane są jakieś problemy

| | MongoDB | Cassandra | HBase | Redis |
|---|---|---|---|---|
| Model | Zorientowany na przechowywanie dokumentów | Zorientowany kolumnowo | Zorientowany kolumnowo | Klucz-wartość |
| API | Zapytania przy pomocy JSON | Java API, Cassandra Query Lanugage | Java API, RESTful HTTP API | RESP - REdis Serialization Protocol |
| Dostępność w razie awarii | Jeśli master ulegnie awarii jego rolę przejmuję jeden ze slave'ów. Trwa to około 10-40 sekund, a przez ten czas zapisywanie do bazy jest niedostępne | Model „multiple master" zapewnia odporność na awarię pojedynczych węzłów | Architektura single master uniemożliwia zapis do bazy w przypadku awarii mastera. | W przypadku awarii mastera wymagane jest ręczne wybranie nowej instancji Awaria mastera uniemożliwia zapis do bazy |
| Przystosowanie do dużej ilości zapisów | W modelu „single master" tylko podstawowy węzeł przyjmuje zapisy, a pozostałe mogą obsługiwać tylko odczyty co istotnie ogranicza skalowalność zapisu | Zapisy może przyjmować którykolwiek z serwerów, a im jest ich więcej tym więcej zapisów może obsłużyć | Podobnie jak Cassandra, ale tutaj log i cache nie są zapisywane jednocześnie, co czyni zapis wolniejszym. | Eliminując potrzebę dostępu do dysków, magazyny danych takie jak Redis pozwalają uniknąć opóźnień w wyszukiwaniu zawartości. Wszystkie dane zostają bowiem zapisywane w pamięci, a nie w przeciwieństwie do standardowych baz danych, na dysku. Redis |

| | | | | |
|---|---|---|---|---|
| | | | | zapewnia czas odpowiedzi liczony w mikrosekundach i umożliwia nawet do kilku milionów żądań na sekundę. |
| Skalowalność dostępności | MongoDB skalowanie odbywa się w sposób poziomy. Zwiększanie możliwości obliczeniowych systemu odbywa się poprzez dodanie nowego urządzenia do klastra | Dostawienie serwera pozwala na lepszą dostępność do bazy | Jest cieńka i zbudowana na małe tabeli. Trudne do skalowania. | Wysoka wydajność i skalowalność |
| Spójność | Domyślnie ciągła, w przypadku umożliwienia odczytu ze slave'ów ewentualna | Ewentualna | Ciągła | Ciągła |
| Twierdzenie CAP (Consistency, Availability, Partition Tolerance) | CP | AP | CP | CP |

Uzasadnienie wyboru systemu baz danych

Po analizie wybranych cech systemów baz danych zdecydowaliśmy się na bazę danych Cassandra. Dla rozważanego problemu najlepszym modelem jest model zorientowany kolumnowo, co eliminuje MongoDB. Redis nie jest dobrym wyborem ponieważ przechowuje dane w pamięci. Przechowywanie w pamięci umożliwia bardzo szybkie odczytywanie i zapisywanie danych, lecz szybkość odczytu nie jest aż tak istotna dla naszego przypadku. Wymogiem jest przechowywanie dużej ilości danych, do czego teoretycznie można zastosować pamięć RAM, lecz spowodowałoby to bardzo wysoki koszt takiego systemu. Z popularnych systemów korzystających z modeli opartych o kolumny wybraliśmy HBase i Cassandra. Oba rozwiązania są szeroko stosowane i dostępnych jest wiele różnych materiałów opisujących ich stosowanie.

Ostatecznie zdecydowaliśmy się na wybór Cassandry. Głównymi powodami tego wyboru były łatwiejsza konfiguracja i lepsza dokumentacja. Konfiguracja Hbase, będącego systemem typu master-slave, za pomocą ZooKeeper'a jest bardziej złożona i może spowodować więcej problemów w przyszłości.

# 2 Prototyp

## 2.1 Prototyp systemu

Po wybraniu systemu baz danych, należy przygotować prototyp systemu. Firma zakłada, że dostęp do bazy danych powinien odbywać się po standardowym RESTful API, tak aby mogły z nim komunikować się różnego typu aplikacje klienckie albo urządzenia.

- Dokładny opis metod RESTful API wymaganych do działania systemu
- Opis wybranej technologii implementacji RESTful API, wraz z informacją o chmurze obliczeniowej, w której rozwiązanie docelowo może zostasć umieszczone. Chmura musi dawać możliwość dystrybucji geograficznej aplikacji i systemu baz danych, tak by RESTful API znajdowało się "możliwie blisko" klienta.

Przygotuj bazę danych. Opis środowiska do zarządzania bazą danych

Procedura instalacji systemu baz danych, skrypt tworzący bazę danych i jej konfigurację

## 2.2 Implementacja RESTful API

Środowisko deweloperskie, narzędzia, język programowania

Wstępna implementacja RESTful API, tak aby móc wykonywać podstawowe operacje CRUD na bazie danych

Listing z podstawowych elementów implementacyjnych

Wykorzystane sterowniki

Parametry połączenia do bazy danych

## 2.3 Środowisko testowe i testy

Przygotuj prosty test obciążeniowy bazy danych poprzez RESTful API bądź inne dostępne narzędzie.

Opis wykorzystanego do tego celu narzędzia oraz konfiguracja testu. Powinny być przetestowane wszystkie operacje CRUD.

Wyniki testów ich analiza i wnioski. Jeżeli testy wykonywane są przez RESTful API, rozdziel czasy obsługi żądań po stronie serwera i po stronie bazy danych

# 3 Skalowanie rozwiązania

## 3.1 Konfiguracja klastra

Zgodnie z wymaganiami, system baz danych musi być skalowalny i zapewniać dużą niezawodność pracy. Przygotuj konfigurację klastra w taki sposób, aby zwiększyć jego wydajność względem pojedynczego systemu baz danych. Minimalna liczba węzłów klastra powinna wynosić 2. Zaleca się sprawdzenie także zachowania systemu dla większej liczby węzłów.

Konfiguracja klastra powinna także uwzględniać możliwą redundancję danych w nim zgromadzonych, tak aby w przypadku "awarii" system działał dalej prawidłowo. Przygotuj procedurę związaną z przywróceniem systemu do pełnego działania po awarii. Oceń skutki awarii poszczególnych elementów systemu na jego działanie. Wyróżnij elementy krytyczne, które uniemożliwią jego działanie.

## 3.2 Wdrożenie

Diagramy wdrożeniowe systemu z opisem wymagań

## 3.3 Konfiguracja Docker i Docker compose

Do przygotowania środowiska zaleca się wykorzystanie obrazów Dockerowych, zarówno dla systemu baz danych jak i dla aplikacji ReST API.

Konfiguracja dla poszczególnych obrazów, wraz z ich modyfikacjami. Wykorzystać Docker compose, który zestawi cały system.

UWAGA! Jeżeli uda się stworzyć klaster Kubernetes/Docker/Grafana/Prometheus i udostępnić go na zajęcia, to będzie to docelowa platforma uruchomieniowa. Problemem mogą być niestety kwestie bezpieczeństwa. Na razie trwają nad tym prace.

## 3.4 Testy

Wykonać test operacji CRUD dla klastra na podobnej zasadzie jak dla pojedynczego węzła. Umieść wyniki testów i porównaj z wcześniejszymi wynikami.

# 4 Rozproszenie geograficzne

## 4.1 Wdrożenie

Diagramy wdrożeniowe systemu wraz z opisem wymagań

Wstępnym założeniem systemu była możliwość jego rozproszenia geograficznego, np. poprzez umieszczenie elementów w chmurze. Oznacza to, że baza danych i aplikacja RESTful API będą znajdowały się w różnych lokalizacjach.

## 4.2 Konfiguracja Docker i Docker compose

Jeżeli uda się wdrożyć Kubernetes, wymagana będzie konfiguracja dla tej platformy.

Zbuduj system w taki sposób aby:

- klaster bazodanowy był rozdystrybuowany na dwa centra danych
- centra danych zapewniały spójność i zgodność danych
- były dostępne dwie instancje aplikacji REStful API, każda z nich skonfigurowana na swoje centrum danych

Ocena czy przy takiej organizacji systemu jest w ogóle możliwe zachowanie spójności danych. Jak parametry konfiguracyjne systemu baz danych wpływają na zachowanie się systemu.

Opis problemów i zjawisk które mogą występować w zależności od konfiguracji.

Przeciwdziałanie niepożądanym zjawiskom.

## 4.3 Testy

Testy dla dwóch niezależnych klientów, z których każdy wykonuje operacje na "swoim" centrum danych. Porównaj wyniki z wcześniej uzyskanymi

# 5 Wnioski

Oceń, czy początkowy wybór systemu baz danych był słuszny. Czy spełnił on stawiane mu wymagania

Czy po wykonaniu projektu, dokonałbyś innego wyboru systemu baz danych

Porównaj swój wybór z możliwościami dostępnych systemów chmurowych i rozwiązań w nich stosowanych

# 6 Literatura

Spis literatury w poprawnym formacie bibliograficznym. Jeżeli nie wiesz jak jest, znajdź stosowną informację na stronie BG PŁ

# 7 Uwagi

1. Pod rysunkami i listingami w sprawozdaniu umieść numerowane podpisy
2. W treści pracy stosuj dodawanie referencji do numerowanych elementów, nie pisz poniżej/powyżej, w kolejnym rozdziale itd.
3. W treści pracy umieść referencje do literatury z której korzystasz np. przy analizie systemów baz danych
4. Przed konwersją do PDF i umieszczeniem sprawozdania na platformie, zaktualizuj spis treści