ROZPROSZONE SYSTEMY OPERACYJNE

PROJEKT

FAZA II

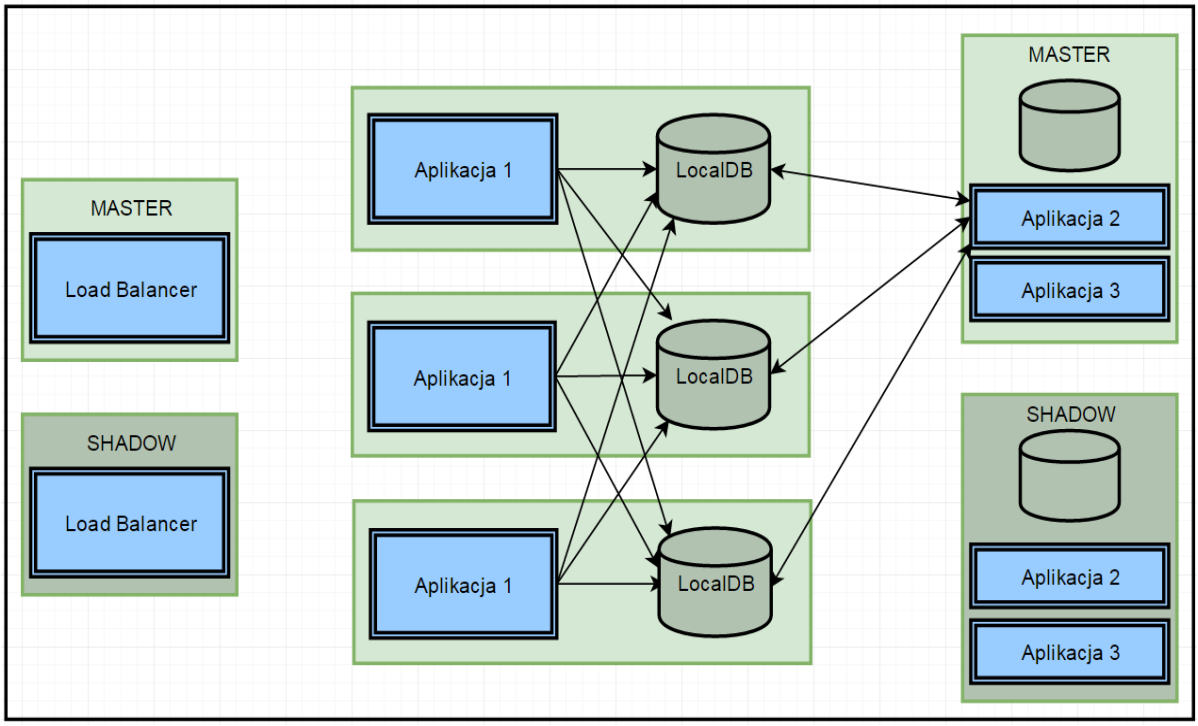
Zespół Mechatroniczni Wojownicy:
Dżumaga Rafał,
Grudzień Ewelina,
Gwiazdowicz Artur,
Jóźwik Mateusz,
Misiowiec Piotr,
Poćwierz Maciej,
Stepek Katarzyna

Politechnika Warszawska

2016-05-27

FAZA II

W ramach fazy II stworzyliśmy prototyp systemu przedstawionego poniżej.



Rysunek 1. Architektura sytemu

Realizowany system dostaje na wejściu dane z głosowania w postaci paczek. Aplikacja 1 wystawia usługę RESTową i otrzymuje dane, które przekierowuje do niej load balancer (LB). Usługa load balancingu wystawiona jest na dwóch węzłach na wypadek awarii jednego z nich. Aplikacja 1 zapisuje dane do bazy MongoDB w sposób redundantny oraz z użyciem zatwierdzania dwufazowego.

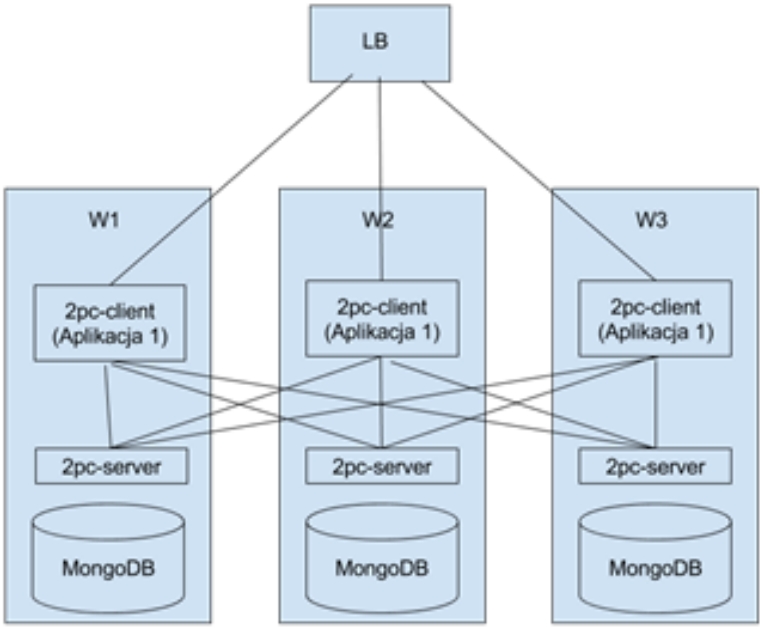
Dane przechowywane w rozproszonej bazie odczytywane są cyklicznie przez aplikację 2, która liczy statystyki i zapisuje wyniki przeprowadzonej analizy do lokalnej bazy PostreSQL.

Aplikacja 3 wystawia usługę umożliwiającą administratorowi danych przeglądanie wyników analizy odczytanych z bazy PostreSQL w przeglądarce. Aby zapewnić dostępność aplikacji 2 i 3 postanowiono uruchomić aplikację 3 jednocześnie na węźle master i shadow. Aplikacja 2 uruchomiona jest w danej chwili tylko na jednym węźle, ale lokalna baza danych dostępna jest na obu węzłach. Jeśli master działa bez problemu, to aplikacja 2 przesyła regularnie do węzła shadow wyniki analizy tak, by obie bazy zawierały ten sam zestaw danych. W przypadku awarii węzła master, aplikacja 2 uruchamiana jest na węźle shadow i kontynuuje cykliczne przetwarzanie. Gdy awaria mastera zakończy się, aplikacja 2 ponownie działa tylko na masterze. Odpytuje ona węzeł shadow o ilość danych w bazie danych i pobiera od niego brakujące dane, jeśli takie istnieją.

W idealnej sytuacji, rozstrzyganie adresów IP domeny, pod którą wystawiamy nasze usługi, realizowane byłoby przez zewnętrzny serwer DNS. Nasz system jest jednak realizowany w uproszczonych realiach, gdzie wszystkie komponenty znajdują się w sieci lokalnej, bez dostępu do internetu, a co za tym idzie serwerów DNS. W tej sytuacji postanowiliśmy zapisać adresy IP, pod którymi znajdują się wystawiane przez nas usługi, w pliku /etc/hosts na komputerze, który z tych usług będzie korzystać.

Udało się nam:

1. Napisać generator danych i aplikację pomocniczą, która wysyła do aplikacji 1 dane w postaci JSON za pomocą zapytań http typu POST. Przykładowy zestaw danych dla pojedynczego głosującego to:
   ```
   {"Pesel":"59111163714","Vote":"Vote1","VotingArea":"VotingArea1",
   "Gender":0, Education":"Education1"}
   ```
2. Utworzyć rozproszoną bazę danych – rozwiązanie składa się z dwóch aplikacji, których instancje działają na wszystkich dostępnych w tym celu węzłach i realizują zatwierdzanie dwufazowe:
   - **2pc-client** - aplikacja udostępnia API REST umożliwiające zapis danych w formacie JSON do bazy rozproszonej. Aplikacja pełni również rolę nadzorcy w protokole zatwierdzania dwufazowego.
   - **2pc-server** - aplikacja pełni rolę podwładnego w protokole 2PC. W procesie 2PC dokonuje bezpośredniego zapisu danych do lokalnej instancji MongoDB.



Rysunek 2. Schemat rozproszonej bazy danych i obsługującej jej aplikacji

3. Stworzyć prototyp aplikacji 2, która zaczytuje dane z pojedynczego węzła z bazą MongoDB, przetwarza je oraz zapisuje wyniki przetwarzania do lokalnej bazy PostreSQL.

   Aplikacja 2 jest napisana w języku Java. Pobiera wstępnie pogrupowane dane z bazy MongoDB, wylicza na ich podstawie statystyki, a zagregowane wyniki zapisuje do bazy PostreSQL.

   Aplikacja 2 będzie działać cyklicznie co zadaną ilość czasu. Na tę chwilę prototyp aplikacji pobiera dane z jednej bazy Mongo. Docelowo obsługiwana będzie większa ilość węzłów z bazami Mongo.

Typy obliczanych statystyk:

- Ile głosów oddano na poszczególne partie w zależności od grupy wiekowej głosujących,
- Ile głosów oddano na poszczególnych kandydatów,
- Ile głosów oddano na partie w poszczególnych okręgach wyborczych,
- Ile głosów oddano na partie w zależności od wykształcenia głosujących,
- Ile głosów oddano na partie ogółem,
- Ile głosów oddano na partie w zależności od płci głosujących.

Stworzona baza PostreSQL przechowuje zagregowane statystyki.

Zawiera tabele słownikowe:

- `d_age` - słownik grup wiekowych głosujących
- `d_parties` - słownik partii
- `d_candidates` - słownik kandydatów z przynależnościami do partii
- `d_constituencies` - słownik okręgów wyborczych
- `d_education` - słownik poziomów wykształcenia głosujących
- `d_sex` - słownik płci

oraz tabele przechowujących zagregowane statystyki:

- `res_party_age` - ilość głosów na partie w zależności od grupy wiekowej głosujących
- `res_party_candidates` - ilość głosów na kandydatów ogółem
- `res_party_constituency` - ilość głosów na partie w zależności od okręgów
- `res_party_education` - ilość głosów na partie w zależności od wykształcenia głosujących
- `res_party_percent` - ilość głosów na partie ogółem
- `res_party_sex` - ilość głosów na partie w zależności od płci głosujących

4. Utworzyć webowa aplikację 3 (dla administratora danych), przy wykorzystaniu frameworku Web2Py. Aplikacja łączy się z bazą PostgreSQL i wyświetla w przeglądarce internetowej wyniki przetworzonych danych w postaci wykresów. Dostępne są następujące statystyki:

   a) frekwencja głosujących
   b) procentowe wyniki głosów w zależności od partii
   c) ilościowe wyniki głosów na partie w zależności od okręgu głosowania (Rys. 3)
   d) procentowy wynik głosowania na kandydatów z danej partii (Rys. 4)
   e) ilościowe wyniki głosów w zależności od wieku
   f) ilościowe wyniki głosów w zależności od wykształcenia
   g) procentowe wyniki głosów na partie w zależności od płci

5. Uruchomić aplikację 1 oraz 3 na Dockerze

6. Przygotować kontenery z aplikacją 1 oraz 3. Kontenery zostały umieszczone w repozytorium DockerHub (repozytorium mateuszj92).

7. Napisać scenariusze testowe. Dostępne w osobnym pliku (scenariuszeTestowe.pdf)

Rysunek 3. Statystyka obrazująca ilościowe wyniki głosów na partie w zależności od okręgu głosowania [Aplikacja 3]

Rysunek 4. Statystyka przedstawiająca procentowy wynik głosowania na kandydatów z danej partii [Aplikacja 3]