

SRDS – Projekt zaliczeniowy

Stworzona przeze mnie aplikacja symuluje kurierów dostarczających paczki do paczkomatów, w celu zaprezentowania rozwiązania problemu współbieżnego dostępu do zasobów w bazie danych Cassandra. Symulowane w aplikacji miasto podzielone jest na dzielnice, w których mieszkają klienci. W każdej dzielnicy znajduje się co najmniej jeden paczkomat. Każdy z paczkomatów posiada z góry określoną pojemność. W centralnym magazynie znajdują się paczki zaadresowane do klientów. Zadaniem kurierów jest pobranie paczek z magazynu i dostarczenie ich do paczkomatów. Każda podróż kuriera rozpoczyna się od przydzielena mu dzielnicy docelowej. Następnie, pobiera on z magazynu pewną liczbę paczek zaadresowanych do klientów mieszkających w tej dzielnicy (maksymalną liczbę pobranych paczek określa pojemność bagażnika), po dotarciu na miejsce, umieszcza je w wybranych przez siebie paczkomatach.

Co istotne, paczki nie są odgórnie przydzielane kurierom, wręcz przeciwnie, to kurierzy sami wybierają, które paczki chcieliby zabrać, zatem kilku kurierów może próbować zabrać z magazynu tę samą paczkę. Miejsca w paczkomatach również nie są odgórnie przydzielane, obowiązek upewnienia się, że w wybranym paczkomacie jest jeszcze miejsce spoczywa więc na kurierze.

Schemat bazy danych

W bazie danych znajdują się 4 tablice, które zawierają podstawowe informacje na temat występujących w aplikacji obiektów. Tablice te służą wyłącznie do odczytu. Są to: **District** (lista dzielnic), **PostBox** (lista paczkomatów), **Courier** (lista kurierów) oraz **Client** (lista klientów).

```
CREATE TABLE District (  
  district text,  
  PRIMARY KEY (district)  
);
```

```
CREATE TABLE Client (  
  client_id text,  
  district text,  
  PRIMARY KEY (client_id)  
);
```

```
CREATE TABLE PostBox (  
  postbox_id text,  
  district text,  
  capacity int,  
  PRIMARY KEY (district, postbox_id)  
);
```

```
CREATE TABLE Courier (  
  courier_id text,  
  capacity int,  
  PRIMARY KEY (courier_id)  
);
```

Poza tym, zdefiniowane zostały również tablice **WarehouseContent** i **PostBoxContent**. Pierwsza reprezentuje aktualną zawartość magazynu, druga wszystkich paczkomatów. Tablica **PostBoxContent** posiada indeks na polu *client_id*, dzięki któremu klienci mogą pobrać listę adresowanych do nich paczek.

```
CREATE TABLE  
PostBoxContent (  
  postbox_id text,  
  package_id text,  
  client_id text,  
  is_ready_to_pickup boolean,  
  PRIMARY KEY (postbox_id,  
  package_id)  
);  
CREATE INDEX  
PostBoxContentClientIndex ON  
PostBoxContent(client_id);
```

```
CREATE TABLE  
WarehouseContent (  
  package_id text,  
  courier_id text,  
  district_dest text,  
  client_id text,  
  PRIMARY KEY (district_dest,  
  package_id)  
);
```

Działanie aplikacji

Pojawienie się w systemie nowej paczki sygnalizowane jest przez dodanie nowego wpisu w tablicy **WarehouseContent**, pola *district_dest* i *client_id* wypełniane są danymi odbiorcy, natomiast pole *courier_id* ma wartość null. Kurier, po dotarciu do magazynu losuje dzielnicę, do której się uda. Następnie pobiera z **WarehouseContent** listę paczek skierowanych do tej dzielnicy, po czym spośród tych, które w polu *courier_id* mają wartość null wybiera paczki, którymi byłby zainteresowany i uaktualnia ich wpisy dopisując w polu *courier_id* swój identyfikator. W celu upewnienia się, że nikt inny nie uprzedził go, kurier czeka przez pewien czas, po czym pobiera z bazy danych informacje na temat zarezerwowanych przed chwilą paczek. Jeśli rekord nie został usunięty z bazy, a pole *courier_id* nadal zawiera jego identyfikator, usuwa rekord z bazy i wkłada paczkę do bagażnika (reprezentowanego przez listę przechowywaną jako pole w obiekcie). Po zebraniu odpowiedniej liczby paczek wyrusza w drogę.

Po dotarciu na miejsce, kurier pobiera z tablicy **PostBox** listę paczkomatów w dzielnicy. Następnie, sprawdza w którym z nich jest wolne miejsce. Dokonuje tego przez porównanie z **PostBoxContent** listy paczek obecnie się w nim znajdujących i porównuje ją z pobraną wcześniej maksymalną pojemnością. Jeśli uzna, że jest jeszcze wolne miejsce, kurier dokonuje rezerwacji miejsca. Rezerwacja polega na wstawieniu do tablicy **PostBoxContent** nowego rekordu z polem *is_redy_to_pickup* ustawionym na false, co znaczy że paczka fizycznie jeszcze nie znajduje się w paczkomacie i klient nie może jej jeszcze odebrać. Rezerwacja jest konieczna, ponieważ inni kurierzy mogą równocześnie próbować włożyć paczki do tego samego paczkomatu, więc podobnie jak w przypadku pobierania paczek z magazynu, kurier nie może mieć pewności czy nikt go nie uprzedził. Podobnie więc, kurier czeka przez chwilę, po czym ponownie pobiera listę paczek w paczkomacie. Jeśli paczkomat okaże się przepełniony, kurier usuwa z niego tyle spośród wstawionych przed chwilą rekordów, by całkowita liczba paczek nie przekraczała maksymalnej pojemności, po czym potwierdza włożenie do paczkomatu paczek poprzez ustawienie pola *is_ready_to_pickup* na true.

Generacji paczek oraz ich odbieraniu przez klientów nie towarzyszą żadne problemy związane ze współbieżnością.

Weryfikacja poprawności

W celu weryfikacji poprawności zaproponowanego rozwiązania problemu współbieżnego dostępu do zasobów, w bazie danych umieszczona została dodatkowa tablica **PackageLog**.

```
CREATE TABLE PackageLog (  
  package_id text,  
  action_type int,  
  action_time timestamp,  
  actor_id text,  
  postbox_id text,  
  PRIMARY KEY (action_time,  
  package_id, actor_id)  
);
```

Umieszczane są w niej wpisy dotyczące faktu wystąpienia pewnych istotnych zdarzeń. Pozwala określić typ wykonanej akcji (*action_type*), kto ją wykonał (pole *actor_id* zawierające id kuriera/klienta), identyfikator obiektu którego dotyczy zdarzenie (*package_id*, *postbox_id*) a także (przybliżony) czas wykonania akcji. Kurierzy odnotowują w niej fakt pobrania paczki z magazynu oraz umieszczenia jej w paczkomacie, a klienci odbiór paczki z paczkomatu.

Na podstawie zawartości tej tablicy można ustalić czy każda z paczek została zabrana z magazynu co najwyżej raz i umieszczona w paczkomacie co najwyżej raz, a także oszacować (niestety, tutaj nie można mieć pewności) czy w żadnym momencie, żaden z paczkomatów nie był przepełniony.