

Zadanie 1: Sieci Liczników SKJ (2017)

Wstęp

Dokładny pomiar upływu czasu jest jednym z ważniejszych zadań w wielu dziedzinach techniki. Synchronizacja oddalonych od siebie liczników jest trudna, w szczególności dlatego, że uwzględnione powinny być opóźnienia przesyłu sygnałów, reakcji urządzeń, itp. W ramach zadania zbudujemy protokół synchronizacji rozproszonej sieci liczników przy użyciu połączeń TCP.

Specyfikacja

Agent

Zarządzaniem wartością liczników zajmują się **agenci**. Każdy z agentów jest osobnym węzłem, identyfikowanym poprzez **adres IP** i **numer portu**. Port ten służy agentowi do komunikacji zarówno z innymi agentami, jak i monitorem (zob. dalej). Każdy z agentów posiada **jeden licznik**, który:

- Odmierza upływający czas w milisekundach.
- Może zostać ustawiony na dowolną wartość.

Ponadto, każdy z agentów ma **pełen obraz sieci** - tzn. zna adresy IP i porty wszystkich pozostałych agentów. Może w tym celu utrzymywać otwarte połączenia, lub bazę danych w postaci książki adresowej. Każdy z agentów może zostać odpytany o bieżącą wartość licznika i sieci poprzez wysłanie następujących komunikatów na **port agenta**:

- **CLK** - agent odsyła wartość licznika w momencie odebrania komunikatu.
- **NET** - agent odsyła numery IP i portów wszystkich znanych agentów.

Interakcje agentów

Dodawanie agentów

Każdy z kolejnych agentów jako parametr otrzymuje wartość początkową swojego licznika (oraz ew. numer portu). Pierwszy agent uruchamiany jest bez dodatkowych parametrów, tworząc pierwszy węzeł sieci i uruchamiając swój licznik. Każdy z kolejnych agentów otrzymuje jako kolejne parametry adres IP i numer portu **agenta wprowadzającego**, już obecnego w sieci.

Po połączeniu ze swoim agentem wprowadzającym, agent pobiera od niego zamiary wszystkich pozostałych. Następnie przekazuje swoje dane (numer IP i port) każdemu z pozostałych agentów. Po dodaniu, agent dokonuje **synchronizacji**, jak opisano w kolejnej sekcji.

Synchronizacja liczników

Każdemu z agentów wydać można polecenie dokonania synchronizacji licznika poprzez wysłanie **komunikatu SYN** na **port agenta**. Przebieg procesu jest następujący:

1. Agent i pobiera od wszystkich agentów wartości ich liczników T_1, \dots, T_N .
2. Agent i ustawia nową wartość swojego licznika na średnią z pobranych wartości (wliczając w to swoją, tzn. T_i):

$$T_i := \frac{1}{N} \sum_{j=1}^N T_j.$$

Ponadto, agent dokonuje synchronizacji w momencie dołączenia do sieci, jak opisano powyżej. Wówczas, po ustawieniu nowej wartości swojego licznika, wysyła do pozostałych agentów komunikat SYN.

Przykład

Rozważmy sieć w której są agenci A i B. Do sieci dołącza agent C, mający agenta wprowadzającego A. Przykładowy przebieg interakcji jest taki:

1. Agent C pobiera od A listę kontaktów.
2. Agent C wysyła swoje dane do A i B, którzy uaktualniają swoje listy kontaktów.
3. Agent C pobiera od A i B wartości liczników i ustawia nową wartość swojego licznika.
4. Agent C wysyła do agentów A i B polecenia SYN.
5. Agent A odbiera komunikat SYN i reaguje, pobierając od B i C wartości liczników i ustawiając nową wartość swojego licznika.
6. Agent B odbiera komunikat SYN i reaguje, pobierając od A i C wartości liczników i ustawiając nową wartość swojego licznika.

Monitor

Monitor jest możliwie prostym programem prezentującym stan sieci agentów. Powinien wyświetlać tabelę zawierającą listę agentów (tzn. par IP:port) wraz z regularnie uaktualnianymi wartościami liczników. Monitor może zostać zrealizowany jako program konsolowy lub (punkty dodatkowe) serwer http będący równocześnie centrum sterowania. W tym ostatnim przypadku, monitor powinien umożliwiać wysyłanie polecenia SYN do dowolnie wybranego agenta (jeden punkt) oraz odłączenie agenta od sieci (jeden punkt). Odłączenie agenta wymaga oczywiście uaktualnienia mapy sieci i ponownej synchronizacji pozostałych agentów.

Wymagania i sposób oceny

1. Poprawny i pełny projekt wart jest **5 punktów** (plus 2 dodatkowe za zrealizowanie opcjonalnej funkcjonalności). Za zrealizowanie każdej z poniższych funkcjonalności można otrzymać punkty do podanej wartości.
 - Poprawne dodawanie agentów do sieci, wymiana danych o połączeniach w sieci. **2 punkty**.
 - Poprawna synchronizacja zegarów. **2 punkt**.
 - Podstawowa funkcjonalność monitora. **1 punkt**.
 - Można uzyskać **dwa dodatkowe punkty** za realizację monitora w postaci serwera http o funkcjonalności jak powyżej.
2. Aplikację piszemy w języku Java zgodnie ze standardem Java 8 (JDK 1.8). Do komunikacji przez sieć można wykorzystać jedynie podstawowe klasy do komunikacji z wykorzystaniem protokołu TCP. Za zgodą prowadzącego grupę można wybrać inny język.
3. Projekty powinny zostać zapisane do odpowiednich katalogów w systemie EDUX w nieprzekraczalnym terminie 26.XI.2017 (termin może zostać zmieniony przez prowadzącego grupę).
4. Spakowany plik projektu powinien obejmować:
 - Plik *Dokumentacja(nr.indeksu)Zad1.pdf*, opisujący, co zostało zrealizowane, co się nie udało, gdzie ewentualnie są błędy, których nie udało się poprawić.
 - Pliki źródłowe (dla JDK 1.8) (włącznie z wszelkimi bibliotekami nie należącymi do standardowej instalacji Javy, których autor użył) - aplikacja musi dać się bez problemu skompilować na komputerach w laboratorium w PJA.

UWAGA: PLIK Z DOKUMENTACJĄ JEST WARUNKIEM KONIECZNYM PRZYJĘCIA PROJEKTU DO OCENY.

5. Prowadzący oceniać będą w pierwszym rzędzie poprawność działania programu i zgodność ze specyfikacją, ale na ocenę wpływać będzie także zgodność wytworzonego oprogramowania z zasadami inżynierii oprogramowania i jakość implementacji.
6. JEŚLI NIE WYSZCZEGÓLNIŁO INACZEJ, WSZYSTKIE NIEJASNOŚCI NALEŻY PRZEDYSKUTOWAĆ Z PROWADZĄCYM ZAJĘCIA POD GROŻBĄ NIEZALICZENIA PROGRAMU W PRZYPADKU ICH NIEWŁAŚCIWEJ INTERPRETACJI.