

Dokumentacja

Zadanie1 Paweł Raszuk s15225

Parametry do uruchomienia programu:

- First Agent: Task1_TCP/src/Main.java agent <value of counter>
np. "Task1_TCP/src/Main.java agent 10001"
- Next Agent: Task1_TCP/src/Main.java agent <value of counter> <ip of agent> <port>
np. "TASK1_TCP/src/Main.java agent 20 10.1.1.2 10001"
- Monitor: Task1_TCP/src/Main.java monitor <ip of agent> <port of agent>
np. "Task1_TCP/src/Main.java monitor 10.1.1.3 10001"

*Program radzi sobie z agentami między maszynami aczkolwiek jeżeli ktoś zacznie stawiać agentów na loopback'u może się wszystko popsuć
także zachęcam do korzystania z jednej sieci i jednego interface'u, czy to loopback czy LAN.

Co zostało zrealizowane:

0. Kod Agentów jest OK myślę, Monitor trochę gorzej. Kwestia Swingowa javy.
Ale całość została okomentowana.

1. Agent inicjalizujący.

- Po wpisaniu parametrów np. agent 20, nasz agent startuje z licznikiem o wartości 20.
- Licznik zapisuje startową wartość z System.currentTimeMillis (startTimeCounter) oraz wartość z parametru(counterOffSet), dzięki temu nie potrzebny jest nieoptymalny i mogący zawierać błędy wątek z pętlą, a z prośbą o CLK wysyłana jest różnica czasów + offset.
- Przyjmuje połączenie do serwera, po czym przerzuca je w nowy wątek traktując jako oddzielnego klienta (multi-client).
- Obsługuje komendę "CLK" odsyłając aktualny stan licznika.

2. Agent (child) bazujący na innym agencie (parent):

- Istnieje książka adresowa zawierająca obiekty typu Contact (ip, port) - obiekty ponieważ był problem z serializacją List w liście.
- Do książki adresowej nie jest dodawany bieżący agent, który z niej korzysta bo po co :D. Dzięki temu omijam sprawdzanie przy wysyłaniu do wszystkich.
- Do dodania kogoś do książki adresowej służy komenda ADD<ip>:<port>
- Analogicznie działa nowa komenda DEL<ip>:<port>
- Zrobiłem to przez Stringa w ten sposób, gdyż wysyłanie wszędzie obiektów nie było by wygodne + inni mają inne referencje do nich

więc mogłyby się pojawić nie potrzebne nowe problemy. IP i port do ADD'a jest dodawany, gdyż agent ma tylko stałe ip i port serwera

Połączeń robi wiele, więc używa wielu socketów co za tym idzie wielu portów oraz ip może zostać potraktowane inaczej, jeżeli wychodzi spoza np. spoza NATu.

- Powstały 2 nowe komendy informujące o opuszczaniu sieci. Komenda OFF powoduje wysłanie do wszystkich z książki adresowej komunikatu DEL który usuwa "tego agenta" książki adresowej, żeby zapobiec nie potrzebnemu wysyłaniu. Następnie stara się zamknąć wszystkie połączenia.

Po czym program powinien zostać zakończony. Tutaj czasami pójdzie exception z killowaniem wątku, aczkolwiek nie wpływa to negatywnie na jakąkolwiek komunikację.

- Została również zastosowana sztuczka. Dodałem komendę RETURN_IP, która zwraca ip agenta wywołującego + zwraca ip agenta nadającego. Przydaje się, gdyż Java domyślnie zwraca 0.0.0.0. Dzięki temu mogę się dowiedzieć jak widzi mnie klient. To daje mi częściową działalność multi-interfaceową.

- Synchronizacja działa między klientami oraz CZEKA na to, aż dany klient odeśle info "DONE", że zsynchronizował swój licznik. Dzięki temu minimalizuję błąd pomiaru.

3. Monitor

- Wyświetla tabele agentów. Ip, port, timer i 2 buttony jeden do wysłania komunikatu SYN, drugi do OFF wyłączenia Agentu.

- Aktualizuje co x sekund tabele

- Aktualizuje dane wysyłając "CLK" do agenta* z listy co x sekund.

* Jeżeli usuniemy tego agenta od którego pobieraliśmy dane. Program powinien autoamtycznie przełączyć nam na kolejnego agenta usuwając poprzedniego z listy. Jeżeli był to ostatni agent powinien nastąpić koniec programu.

- Można ustawić czas odświeżania.

Notka: Testowałem działanie przy 20 agentach. Program działa bez zarzutów. Przy 4 udało mi się ich zsynchronizować do 0-1ms różnicy.

Co nie zostało zrealizowane + ewentualne błędy:

- GUI mogło zostać zaprojektowane lepiej, ale funkcjonalność działa. Jedynie mogło być ładniejsze.

- GUI Mogło być bardziej „odporne na użytkowników” tzn. jak mam okienko z wpisaniem inta blokować to na inne typy np. String.

- Lepsza obsługa błędów, jeżeli coś zabije proces agenta. To GUI wariuje i agenci mogą rzucać exceptionami. Nie spowoduje zatrzymania ich pracy. Aczkolwiek na GUI w tabelce będzie Agent Widmo i wartość może się zmieniać np. od -1 do 999 jeżeli zostawimy domyślne czasy.

- Czasami przy wyłączaniu agentów pojawi się exception z killowania wątku, ale nie wpływa on negatywnie na komunikacje agentów/monitora między sobą, więc jest to raczej tylko niepożądany efekt wizualny.

- Interface'y myślę, że to zbyt obszerny temat jak na ten moment. Działa jak wspomniałem wyżej w części przypadków. Nie byłem w stanie sprawdzić dla których nie działa, gdyż nie wiem jak to testować + nie było by to łatwe.

- Czystszy kod, gdzieś indziej :).

Komendy:

- ADD<ip:port> - nic nie zwraca. Dodaje ip do książki adresowej jako nowy obiekt klasy Contact.

- CLK – zwraca licznik innego agenta.

- DEL<ip:port> - nic nie zwraca. Usuwa ip z książki adresowej jako obiekt klasy Contact.

- NET – zwraca książkę adresową w postaci HashSet<Contact>.

- OFF – nic nie zwraca. Kończy połączenia wywołuje usunięcie go z listy u wszystkich po czym wyłącza agenta.

- RETURN_IP - zwraca ip. Po czym oczekuje IP od agenta. Chodzi o to, żeby zobaczyć jakie ma się „zewnętrzne ip” i to działa ponieważ.

- SYN – zwraca DONE jak skończy. Synchronizuję swój czas pobierając czasy od agentów.

W razie jakichkolwiek wątpliwości, proszę pytać. Podejrzewam, że część odpowiedzi można znaleźć w komentarzach w kodzie.