

Data: 10.12.2016

Marek Gadzalski 191422

Piotr Kowalski 189702

Sprawozdanie z zadań 7-8

1. Cel zadania

Celem zadania był opracowanie aplikacji umożliwiającej wykonanie obliczeń rozproszonych na wielu komputerach. Zadanie obejmowało mnożenie trzech macierzy liczb rzeczywistych o rozmiarach 1024×1024 .

2. Architektura aplikacji

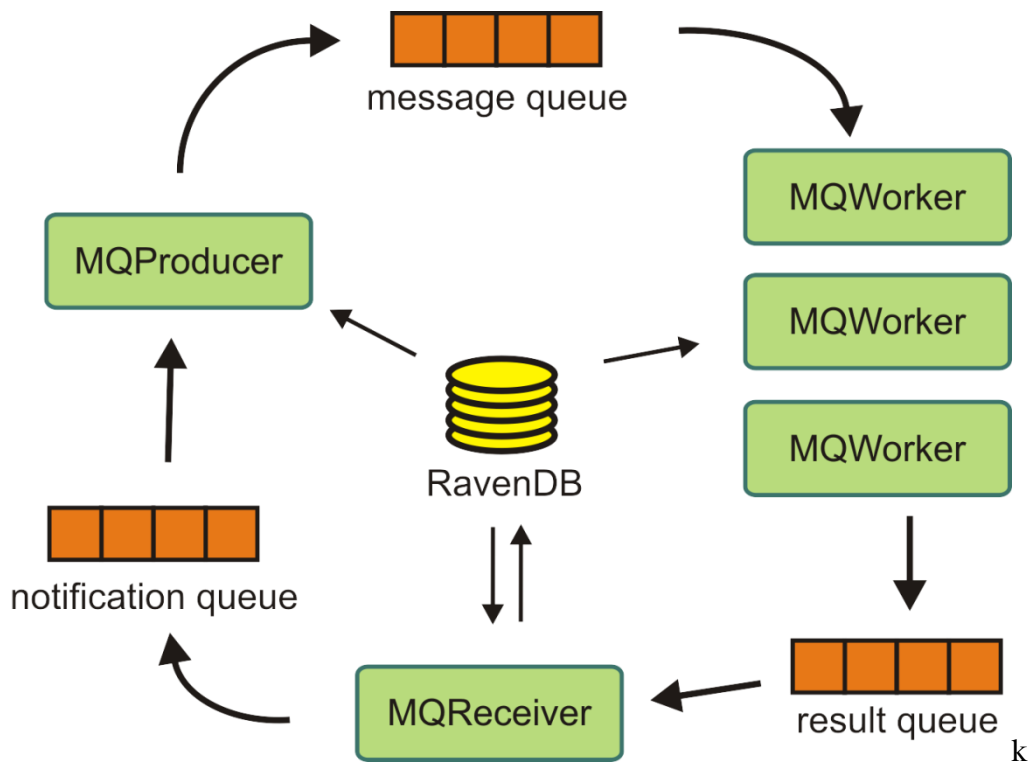
Proces mnożenia tablic podzielny został na jednostki pracy (ang. *Unit of Work* **UOW**) obejmujące mnożenie pojedynczego wiersza z całą drugą macierzą. Zatem mnożenie dwóch macierzy o rozmiarach 1024×1024 złożone jest z 1024 jednostek pracy. Wynikowe wektory są następnie składane do macierzy wynikowej.

Za podział działań na UOW odpowiada pierwszy komponent aplikacji **MQProducer**. Proces ten na podstawie rozmiarów macierzy dzieli ją na UOW i przekazuje dalej. UOW odbierane są przez drugi element aplikacji **MQWorker**, proces ten odbiera UOW i wykonuje odpowiednie obliczenia. Wynik pracy przekazywane jest do trzeciego komponentu **MQReceiver**, który zbiera wyniki działań i składa je do kompletnej macierzy wynikowej.

Komunikacja pomiędzy procesami zapewniona jest poprzez użycie kolejki wiadomości (message queue MQ). W tym celu wykorzystaliśmy serwer **RabbitMQ**. Wykorzystanie mechanizmu kolejek rozwiązuje problem komunikacji i synchronizacji procesów. Eliminuje

również konieczność zapewnienia bezpośredniej łączności pomiędzy komponentami, dzięki temu do systemu można wpiąć dowolną liczbę procesów MQWorker. Co więcej procesy obliczeniowe mogą być dołączane i odłączane w trakcie pracy całego systemu bez zaburzania jego działania.

Aby zminimalizować wielkość danych przesyłanych poprzez sieć, do przechowywania macierzy zastosowano bazę danych. W tym celu wykorzystano bazę danych **RavenDB**. Baza ta jest bazą danych NoSQL i przechowuje dane w postaci dokumentów json.



Rysunek 1 Diagram architektury aplikacji i przepływu informacji

3. Opis działania aplikacji

Aby zapewnić prawidłową pracę aplikacji należy uruchomić i skonfigurować serwery **RabbitMQ** oraz **RavenDB**. Przy czym serwery mogą pracować na innych komputerach niż sama aplikacja, pod warunkiem zapewnienia komunikacji protokołem TCP/IP.

Przed uruchomieniem właściwej aplikacji należy przygotować dane wejściowe. Do tego celu służy aplikacja **DataGenerator.exe**. Aplikacja ta generuje trzy tablice o zadanej wielkości i wypełnia je losowymi liczbami rzeczywistymi, następnie zapisuje te tablice w bazie danych. Następnie uruchamiane trzy składowe elementy aplikacji (**MQWorker**, **MQProducer**, **MQReceiver**). Każdy z tych procesów po uruchomieniu nawiązuje połączenie z bazą danych, z której pobiera macierze wejściowe oraz łączy się z serwerem **RabbitMQ**.

Proces obliczeń inicjowany jest przez proces **MQProducer**. Na podstawie wielkości macierzy, producent generuje UOW i wysyła je na kolejkę **message queue**. Wszystkie podłączone procesy **MQWorker** pobierają po jednej wiadomości UOW i wykonują

odpowiednie obliczenia. Po tym wysyłają wynik na kolejkę **result queue**. Po poprawnie zakończonej operacji wysyłane jest potwierdzenie do MQProducer, który w razie nie otrzymania tegoż wysyła ponownie odpowiedni UOW. Jak wcześniej zaznaczono, procesów MQWorker może być wiele i mogą one być podłączane i odłączane w trakcie obliczeń.

Proces **MQReceiver** pobiera z kolejki result queue wyniki obliczeń, w chwili gdy otrzyma ostatni wynik, składa wynikową macierz i zapisuje je do bazy danych oraz informuje proces MQProducer o zakończeniu pierwszego etapu pracy. W pierwszej pętli działania aplikacji zapisywany jest wynik mnożenia dwóch pierwszych macierzy, natomiast zadanie zakłada mnożenie trzech. Zatem uruchamiana jest druga pętla działania aplikacji, która ma podobny przebieg z tą różnicą, że procesy pobierają z bazy danych tablicę wynikową z pierwszej pętli.

4. Wyniki

W celu przetestowania pracy aplikacji wykonano szereg obliczeń na jednym lub dwóch komputerach połączonych siecią. Serwery MQ oraz RavenDB znajdowały na komputerze z procesorem Intel i7 4790k.

Wielkość macierzy	1024x1024
Liczba procesów MQWorker	Intel i7 4790k: x1
Ilość danych przesyłana kolejkami	1024 UOW · 5 b = 5120 B 1024 Result · 10749 = 11006976 B Łącznie: 11 MB
Ilość danych odczytywanych z bazy danych	MQProducer = 45 MB MQWorker = 45MB MQReceiver = 45MB Łącznie = 135 MB
Obciążenie MQWorker	id: 1, 2048 obliczeń średnio 12 ms
Całkowity czas pracy	50125 ms

Wielkość macierzy	1024x1024
Liczba procesów MQWorker	Intel i5 3210M: x1
Ilość danych przesyłana kolejkami	1024 UOW · 5 b = 5120 B 1024 Result · 10749 = 11006976 B Łącznie: 11 MB
Ilość danych odczytywanych z bazy danych	MQProducer = 45 MB MQWorker = 45MB MQReceiver = 45MB Łącznie = 135 MB
Obciążenie MQWorker	id: 1, 2048 obliczeń średnio 15 ms
Całkowity czas pracy	55125 ms

Wielkość macierzy	1024x1024
Liczba procesów MQWorker	Intel i7 4790k: x2
Ilość danych przesyłana kolejkami	1024 UOW · 5 b = 5120 B 1024 Result · 10749 = 11006976 B Łącznie: 11 MB
Ilość danych odczytywanych z bazy danych	MQProducer = 45 MB MQWorker = 90MB MQReceiver = 45MB Łącznie = 180 MB
Obciążenie MQWorker	id: 1, 1024 obliczeń średnio 12 ms id: 2, 1024 obliczeń średnio 13 ms
Całkowity czas pracy	34246 ms

Wielkość macierzy	1024x1024
Liczba procesów MQWorker	Intel i5 3210M: x2
Ilość danych przesyłana kolejkami	1024 UOW · 5 b = 5120 B 1024 Result · 10749 = 11006976 B Łącznie: 11 MB
Ilość danych odczytywanych z bazy danych	MQProducer = 45 MB MQWorker = 90MB MQReceiver = 45MB Łącznie = 180 MB
Obciążenie MQWorker	id: 1, 1038 obliczeń średnio 18 ms id: 2, 1010 obliczeń średnio 19 ms
Całkowity czas pracy	40878 ms

Wielkość macierzy	1024x1024
Liczba procesów MQWorker	Intel i7 4790k: x1 Intel i5 3210M: x1
Ilość danych przesyłana kolejkami	1024 UOW · 5 b = 5120 B 1024 Result · 10749 = 11006976 B Łącznie: 11 MB
Ilość danych odczytywanych z bazy danych	MQProducer = 45 MB MQWorker = 90MB MQReceiver = 45MB Łącznie = 180 MB
Obciążenie MQWorker	id: 1, 1210 obliczeń średnio 11 ms

	id: 2, 838 obliczeń średnio 15 ms
Całkowity czas pracy	31 972 ms

Wielkość macierzy	1024x1024
Liczba procesów MQWorker	Intel i7 4790k: x2 Intel i5 3210M: x2
Ilość danych przesyłana kolejkami	1024 UOW · 5 b = 5120 B 1024 Result · 10749 = 11006976 B Łącznie: 11 MB
Ilość danych odczytywanych z bazy danych	MQProducer = 45 MB MQWorker = 180 MB MQReceiver = 45MB Łącznie = 270 MB
Obciążenie MQWorker	id: 1, 685 obliczeń średnio 13 ms id: 2, 694 obliczeń średnio 12 ms id: 3, 326 obliczeń średnio 20 ms id: 3, 343 obliczeń średnio 19 ms
Całkowity czas pracy	26 220 ms