# Sprawozdanie z laboratorium:
# Przetwarzanie Rozproszone

Rozwiązanie zadania Modele XXL z użyciem środowiska MPI

13 czerwca 2017

Prowadzący: dr inż. Arkadiusz Danilecki


| | | | | |
|---|---|---|---|---|
| Autorzy: | **Sebastian Firlik** | inf122485 | I2 | sebastian.firlik@student.put.poznan.pl |
| | **Michał Wójcik** | inf122513 | I2 | michal.p.wojcik@student.put.poznan.pl |


Zajęcia wtorkowe nieparzysty tydzień, 15:10.

Oświadczamy, że niniejsze sprawozdanie zostało przygotowane wyłącznie przez powyższych autorów, a wszystkie elementy pochodzące z innych źródeł zostały odpowiednio zaznaczone i są cytowane w bibliografii.

# 1 Wstęp

Naszym zadaniem w ramach projektu z laboratorium z przedmiotu Przetwarzanie Rozproszone było:

- opracować algorytm, rozwiązujący zadany problem,
- zaimplementować go np. z użyciem języka C++ i środowiska Open MPI,
- stworzyć sprawozdanie, pozwalające na samodzielne zaimplementowanie tego algorytmu.

Temat, który został nam przydzielony to:

Modele XXL

> Czas skończyć ze stereotypem, że mężczyzna, by był piękny, musiał być wysportowany, wysoki i szczupły. Mężczyźni nie są obiektami seksualnymi! Ruch feministyczny postanowił więc urządzić konkursy rozlazłych, tłustych, niskich kurdupli, by w ten sposób promować postawę, że o wartości mężczyzny decyduje jego wnętrze, a nie sześciopak na brzuchu.
>
> Proces-agent reprezentuje jednego modela. Co pewien czas procesy chcą zorganizować konkurs. Nie wszystkie procesy mogą chcieć uczestniczyć, niektóre mogą właśnie uczestniczyć w innym konkursie. Do zorganizowania konkursu wymagane jest zarezerwowanie jednej z sal w mieście; istnieje niewiele M miast, w każdym z nich jest kilka S sal. W każdym mieście jest też jeden hotel o X miejscach. Każdy proces uczestniczący w konkursie w danym mieście musi następnie samodzielnie zarezerwować miejsce w odpowiednim hotelu. Po zakończeniu konkursu, sala zwalniana jest natychmiast, a hotel dopiero po pewnym czasie.

W następnych rozdziałach przedstawimy kolejno ideę algorytmu (Rozdział 2), następnie analizy złożoności komunikacyjnej i czasowej, wraz z wszelkimi założeniami dotyczącymi środowiska komunikacyjnego (Rozdział 3).

# 2 Opis zastosowanego algorytmu

Nasze rozwiązanie opiera się zasadniczo na dwóch scenariuszach i je trzeba osobno nakreślić i oddzielić od siebie. Każdy proces może wylosować (czy też wybrać) jedną z dwóch roli. Może być albo uczestnikiem konkursu, albo jego organizatorem.

Przed wybraniem jakiejkolwiek roli jednakże, należy poczynić pewne kroki, by ją ustalić. Po pierwsze, na samym początku każdego cyklu wysyłamy do wszystkich pozostałych procesów-agentów zapytanie o to, czy organizują już konkurs. Jeśli choć jedna spośród N-1 (gdzie N to liczba procesów-agentów) odpowiedzi będzie brzmiała "tak, w mieście $M_1$" to uruchamiamy losową funkcję GenerujRolę(), której celem jest z pewnym rozkładem prawdopodobieństwa (u nas 20% szans na bycie organizatorem, 80% szans na bycie uczestnikiem) wylosowanie jednej z dwóch dostępnych ról.

Jeśli zostaniemy uczestnikiem konkursu to wybieramy pierwszy przybyły do nas konkurs (w wiadomości otrzymaliśmy identyfikator procesu nadawcy) jako ten, w którym zechcemy wziąć udział.

Jeśli wylosujemy rolę Organizatora to przechodzimy do podrozdziału "Rola organizatora konkursu" i kontynuujemy algorytm.

W przypadku gdy wszystkie N-1 odpowiedzi będą przeczące ("Nie, nie organizuję konkursu") to znaczy, że my musimy zorganizować, by nie doprowadzić do utworzenia samych uczestników konkursów przy braku jakiegokolwiek konkursu do wyboru.

W międzyczasie inny wątek odpowiada przecząco na przybyłe pytania "Czy organizujesz konkurs?", jako iż jeszcze nie wybrał żadnej roli lub nie mamy możliwości przechowywania listy uczestników (w podrozdziale o roli organizatora będzie to wyjaśnione).

## 2.1 Rola uczestnika konkursu

Jeśli udało nam się wylosować rolę uczestnika to mamy już zapisane ID organizatora konkursu, w którym bierzemy udział i ID miasta, w którym ten konkurs się odbędzie. Pozostałym ewentualnym organizatorom odpowiadamy na bieżąco, że nie bierzemy udziału w ich konkursie. Teraz musimy poczekać na zakończenie zapisów do naszego konkursu – organizator wyśle nam stosownie otagowaną wiadomość, która będzie nam mówiła, że zapisy się skończyły.

W tym momencie zaczynamy się ubiegać o sekcję krytyczną, którym jest hotel w mieście $M_1$, które otrzymaliśmy w wiadomości od organizatora konkursu. Jeśli otrzymamy N - X wiadomości z pozwoleniami to możemy zająć miejsce w hotelu. Jeśli przyjdzie do nas pytanie o nasz hotel to odsyłamy zgodę jedynie, gdy zegar nadawcy jest wcześniejszy niż nasz (ubiegał się wcześniej niż my, on ma priorytet). Jeśli ubiegał się później niż my to zapisujemy jego identyfikator na listę identyfikatorów do obsłużenia po zwolnieniu sekcji krytycznej. W przypadku zapytania o hotel w innym mieście, zawsze odsyłamy zgodę.

Gdy tylko otrzymamy miejsce w hotelu w mieście $M_1$ to wysyłamy organizatorowi naszego konkursu informację o tym, że już na pewno bierzemy udział i jesteśmy gotowi na rozpoczęcie konkursu.

Od tego czasu oczekujemy na kolejną wiadomość od organizatora ("Zakończyłem konkurs") i wtedy po pewnym czasie zwalniamy hotel i wszystkim osobom, które pytały o nasz hotel w mieście $M_1$ w czasie, gdy my go zajmowaliśmy.

Po tym proces-agent wraca na początek algorytmu – na nowo nie ma roli, będzie próbował losować rolę i kontynuować pracę.

## 2.2 Rola organizatora konkursu

Jeśli rola procesu-agenta zostaje ustalona na organizatora konkursu to proces ten musi wylosować sobie numer miasta i numer sali w nim. Wtedy zaczynamy starać się o wybraną salę w wybranym mieście. Wysyłamy do wszystkich pytanie o to, czy możemy zająć tę salę i czekamy o N-1 zgód. Jeśli dostaniemy pytanie od kogoś innego to zależnie od wartości jego zegara, przechowujemy go w kolejce osób, którym odpowiemy po zwolnieniu sali (jeśli ma zegar późniejszy) lub odsyłamy zgodę (jeśli ma zegar wcześniejszy). Każdy proces odsyła zgody automatycznie, jeśli nie dotyczy to jego sali (czyli inna sala w naszym mieście, sala w innym mieście).

W tym momencie wyślij do wszystkich wiadomość o tym, że organizujesz konkurs i dodaj na listę potencjalnych uczestników wszystkie procesy poza sobą (N-1).

Teraz organizator będzie otrzymywać kilka typów wiadomości.

- Jeśli otrzymamy pytanie "Czy organizujesz konkurs?" to jeśli mamy miejsce możliwe to odpowiadamy, że tak i ewentualnie dopisujemy ten proces na listę potencjalnych uczestników (możliwe, że został usunięty bo wcześniej odpowiedział na nasze zaproszenie przecząco, ale teraz zmienił zdanie).
- Jeśli otrzymamy odpowiedź "Nie biorę udziału" to wykreślamy ten proces z listy potencjalnych uczestników.
- Jeśli otrzymamy odpowiedź "Biorę udział" to wykreślamy go z listy potencjalnych uczestników, a dodajemy na listę pewnych uczestników. Jeśli lista potencjalnych uczestników jest pusta (na nikogo nie czekamy już, wszyscy odmówili potwierdzili, że chcą

brać udział) to zamykamy zapisy. Wtedy rozsyłamy wszystkim zgodę na uzyskiwanie hotelu.

Jeśli otrzymamy od wszystkich pewnych uczestników potwierdzenie, że załatwili sobie miejsca w hotelu to startujemy zawody, wysyłamy wszystkim uczestnikom potwierdzenie zakończenia konkursu i agent wraca do początku algorytmu.

# 3 Analiza złożoności i założeń

Pod uwagę bierzemy dwie złożoności:

- Złożoność czasowa – jest to liczba kroków algorytmu do czasu jego zakończenia, czas przesyłania jest jednostkowy, czas przetwarzania lokalnego zaniedbywalnie mały.
- Złożoność komunikacyjna – ile wysyłamy pakietów / bajtów, my mierzymy złożoność pakietową.

Zgodnie z założeniami złożoności czasowej, zakładamy, że kroki sa wykonywane synchronicznie.

Zależnie od roli, złożoności te będą się od siebie różnić.

## 3.1 Złożoności czasowa i komunikacyjna dla uczestnika

W algorytmie są następujące kroki:

1. Wyślij do N-1 procesów pytanie o to, czy organizują jakiś konkurs
2. Otrzymujemy N-1 odpowiedzi "Tak" lub "Nie" (minimalnie jedna "Tak", byśmy mogli wylosować rolę uczestnika – to będzie konkurs, w którym weźmiemy udział)
3. Wysyłamy jedną wiadomość do Organizatora, że weźmiemy udział w jego konkursie
4. Otrzymujemy jedną wiadomość od Organizatora naszego konkursu, że zapisy się skończyły i mamy zacząć załatwiać miejsce w hotelu
5. Wysyłamy N-1 wiadomości z pytaniem o miejsce w hotelu
6. Gdy otrzymamy N-X zgód to zabieramy miejsce, pozostałe X-1 wiadomości też do nas dotrze na pewno
7. Wysyłamy organizatorowi jedną wiadomość z potwierdzeniem zajęcia hotelu
8. Otrzymujemy jedną wiadomość o końcu konkursu

Złożoność czasowa wynosi więc 8, tyle ile jest kroków algorytmu (zakładamy, że broadcast jest wykonywany optymalnie, równolegle)

Złożoność komunikacyjna wynosi $4 * (N - 1) + 4$, wszelkie dodatkowe, asynchroniczne, przybywające w dowolnym momencie wiadomości nie liczą się do złożoności, ponieważ może ich być potencjalnie nieskończenie wiele, a nie stanowią części algorytmu od strony procesu, który obecnie rozpatrujemy.

## 3.2 Złożoności czasowa i komunikacyjna dla organizatora

Od strony organizatora przebieg algorytmu wygląda następująco:

1. Wysyłamy N-1 pytań o to, czy proces organizuje konkurs
2. Otrzymujemy N-1 odpowiedzi "Tak" lub "Nie"
3. Wysyłamy N-1 pytań o to, czy możemy zająć salę $S_1$ w mieście $M_1$
4. Otrzymujemy N-1 odpowiedzi (zgód), więc zajmujemy salę
5. Wysyłamy X zaproszeń do wzięcia udziału w naszym konkursie
6. Otrzymujemy X odpowiedzi "Biorę udział" lub "Nie biorę udziału"
7. Wysyłamy X zgód na szukanie miejsca w hotelu
8. Otrzymujemy X potwierdzeń, że uczestnik ma już zajęty hotel
9. Wysyłamy X informacji o zakończeniu konkursu

Złożoność czasowa wynosi 9, tyle jest kroków w naszym algorytmie od strony organizatora od początku do końca.

Złożoność komunikacyjna wynosi $4 * (N - 1) + 5 * X$

## 3.3 Założenia środowiska komunikacyjnego

W żadnym momencie działania programu nie potrzebujemy, by pojedynczy kanał był pojemniejszy niż 3 wiadomości, lecz najczęściej jest mniej wiadomości w kanale. Zakładamy, że 3 wiadomości mogą znaleźć się jednocześnie w kanale.