

Rozproszone Systemy Informatyczne

Równoległy algorytm Prima do znajdowania MST

Patryk Koryzna

Wydział Informatyki i Zarządzania, Politechnika Wrocławska

Semestr letni 2014/15

1 Opis algorytmu

Algorytm Prima służy do wyznaczania minimalnego drzewa rozpinającego MST (*ang. minimal spanning tree*), to jest drzewa zawierającego wszystkie wierzchołki grafu ważonego, połączone krawędziami o najniższych wagach.

1.1 Wersja standardowa

Standardowa wersja algorytmu Prima wygląda następująco:

1. Wybierz dowolny wierzchołek v_0
2. Dodaj do drzewa krawędź o najniższej wadze z v_0 do dowolnego innego wierzchołka
3. Z krawędzi wszystkich wierzchołków już znajdujących się w drzewie do wierzchołków jeszcze się w nim nie znajdujących wybierz ten, który ma najniższą wagę i dodaj do MST
4. Powtarzaj 3, dopóki nie dodane zostaną wszystkie wierzchołki grafu

1.2 Wersja równoległa

Krok 3 wersji standardowej można zrównoleglić, zakładając użycie reprezentacji grafu jako macierzy sąsiedztwa. Rozdzielając (w jednym wymiarze) macierz na *partycje*, krok wyszukiwania krawędzi można przeprowadzić współbieżnie.

1.3 Modyfikacja szeregowego algorytmu

1. Wybierz dowolny wierzchołek v_0
2. Dodaj do drzewa krawędź o najniższej wadze z v_0 do dowolnego innego wierzchołka
3. Rozdziel macierz sąsiedztwa na n partycji, gdzie n to ilość węzłów prowadzących obliczenia równoległe (tutaj: aktorów)
4. Dla każdej partycji:

- (a) Wyślij partycję i aktualne wierzchołki grafu do oddzielnego węzła przetwarzania z głównego aktora
 - (b) Z krawędzi ze wszystkich wierzchołków w danej partycji, już znajdujących się w drzewie, do wierzchołków jeszcze się w nim nie znajdujących wybierz ten, który ma najniższą wagę i wyślij do głównego aktora
 - (c) W głównym aktrze odbierz wyniki dla każdej partycji
5. Wybierz z odebranych krawędzi tę o najniższej wadze i dodaj ją do MST
 6. Powtarzaj 4, dopóki nie dodane zostaną wszystkie wierzchołki grafu

2 Implementacja

Do implementacji został wykorzystany język programowania **Scala** (kompilowany do bytecode JVM). Jako abstrakcja obliczeń równoległych został model aktorowy, implementowany w bibliotece **Akka**. Dodatkowo, do koordynacji połączeń między różnymi fizycznymi maszynami został wykorzystany moduł **Akka Clustering**.

2.1 Uruchamianie

Do skompilowania potrzebny jest **SBT**¹. Workery uruchamiamy poleceniami:

- `sbt 'runMain rsi.graph.distributed.actors.BackendWorker 2551'` dla pierwszej instancji
- `sbt 'runMain rsi.graph.distributed.actors.BackendWorker 2551'` dla pozostałych

Frontend wysyłający do workerów wiadomości uruchamiany jest poleceniem `sbt 'runMain rsi.graph.distributed.actors.PrimFrontend 2552'`

¹<http://www.scala-sbt.org/download.html>