Streszczenie

W obecnym świecie, nie tylko w bardzo szybkim tempie rośnie ilość danych, ale również stają się one coraz bardziej połączone, dlatego rozwój narzędzi i nowych podejść do rozwiązywania problemów grafowych cieszy się niesłabnącym zainteresowaniem.

Odnosząc się do wspomnianych wyżej kwestii, sporządzono równoległe opracowania wybranych algorytmów grafowych (Benchmark Graph500) w modelu PGAS dla języka Java. Ideą modelu PGAS jest poprawa efektywności programistów przy jednoczesnym zachowaniu wysokiej wydajności obliczeń. Model PGAS łączy w sobie założenia modelu przesyłania komunikatów (MPI, ang. *Message Passing Interface*) oraz modelu z pamięcią wspólną. Opracowane rozwiązanie zostało zaprojektowane i zaimplementowane w języku Java, który obecnie odgrywa znaczącą rolę przy analizie danych. Jednak, w tradycyjnych wielkoskalowych obliczeniach (HPC) nieprzerwaną uwagą cieszą się takie języki jak Fortran, C/C++ bazujące na modelu przesyłania komunikatów przy komunikacji pomiędzy różnymi węzłami oraz na modelu pamięci wspólnej dla komunikacji wewnątrz jednego węzła [1]. Obliczenia przeprowadzone zostały przy użyciu biblioteki PCJ. Kod opracowania, wraz z pomocniczymi testami opisanymi w niniejszej pracy, jest udostępniony w repozytorium Git.

Pracując nad wydajnością oraz skalowalnością rozwiązania opracowano między innymi algorytmy pozwalające na ograniczenie wymiany informacji, nakładanie obliczeń i komunikacji, co było możliwe dzięki komunikacji jednostronnej charakterystycznej dla modelu PGAS. Dokonano również technicznych badań np. na temat wykorzystywanych struktur danych, biorąc pod uwagę specyfikę języka Java, co pozwoliło na osiągnięcie lepszych wyników. Opracowane rozwiązana zweryfikowano pod względem skalowalności oraz wydajności na różnych architekturach sprzętowych i porównano z rozwiązaniem *state-of-the-art* w języku C z wykorzystaniem biblioteki MPI. Opracowanie pozwoliło na weryfikację przydatności modelu PGAS przy użyciu biblioteki PCJ w języku Java do analizy dużych danych grafowych.

Przedstawione w pracy wyniki testów wydajnościowych pokazują, że obliczenia równoległe i rozproszone na danych grafowych wykonane w modelu PGAS w języku Java z użyciem biblioteki PCJ, pozwalają na uzyskanie dobrej wydajności i skalowalności. Kluczowe jest jednak zapewnienie szybkiego łącza między węzłami klastra oraz wystarczająco dużej ilości pamięci na węzłach. Opracowane rozwiązania oraz uzyskane wyniki świadczą o możliwym wykorzystaniu modelu PGAS i języka Java do efektywnego przetwarzania danych grafowych przy analizie danych, opartej o języki wysokiego poziomu. Jest to szczególnie ważne nie tylko ze względu na osiągnięcie zadowalającej wydajności i skalowalności, ale również trudności w integracji typowego środowiska uruchomieniowego np. systemu kolejkowego, instalacji bibliotek do analizy danych na klastrach itp.. Wykorzystanie czystego języka Java i modelu PGAS dla algorytmów grafowych otwiera możliwości prostego zastosowania w obszarze analizy dużych danych grafowych (*Big Data*).

Abstract

Coverage of selected, parallel graph algorithms in PGAS model and their implementation in Java language

In the contemporary world, not only is the amount of data increasing rapidly, but also it is becoming more and more connected, hence the advancement of tools and new attitudes towards the way of solving graph problems is attracting an anabated interest.

To address the aforementioned issues selected parallel and distributed graph algorithms (Benchmark Graph500) in the PGAS model and in Java language have been developed. The idea of the PGAS model is to increase the programmers' efficency while at the same time preserving high computation performance. PGAS model combines the features of the message passing model and the shared memory model. The developed algorithms were designed and implemented in the Java language, which currently plays a significant role in data analysis. However, in the traditional high performance computations (HPC) languages such as Fortrant, C/C++, based on message passing model in inter-node communication and shared memory model in intra-node communication, draw unwavering attention [1]. Calculations have been carried out using PCJ library. The source code, together with supplementary tests described in this dissertation, are available in the Git repository.

While working on performance and scalability of the proposed solution, various methods for limiting comunitation exchange, overlapping computation and communication have been used. It was possible thanks to one-sided communication characteristic for PGAS model. Moreover, some technical studies have been carried out on used data structures, taking into consideration specificity of Java language, which allowed to achieve a better performance. The proposed solutions were verified in case of scalability and performane on different hardware architectures and compared with *state-of-the-art* solution in C langauage and MPI. The approach used helped to verify the usefullness of the PGAS model, PCJ library and the Java language to analyse big graph data.

Performance tests results shown in the dissertation, prove that parallel and distributed computations on graphs carried out using PGAS model and the Java language, allow to gain good performance and scalability. However, the crucial part is to provide fast link between cluster nodes and sufficient amount of available memory on nodes. The solutions and the results that have been worked out show that PGAS model together with the Java language can be used in effective processing of graphs in data analysis, based on high-level programming languages. This is especially important not only because of gaining good performance and scalability but also due to possible difficulties in integration of typical launch environment e.g. queueing systems, installation of libraries for data analysis on clusters etc. Usage of pure Java language and PGAS model for graph algorithms opens possibilities for easy application in the field of big data analysis.