学際大規模情報基盤共同利用・共同研究拠点 公募型共同研究 平成29年度採択課題
jh170049-ISJ
9th Symposium
Joint Usage / Research Center for Interdisciplinary Large-scale Information Infrastructures

Hiroaki Kobayashi (Graduate School of Information Sciences, Tohoku University)

Theory and Practice of Vector Processing for Data and Memory Centric Applications

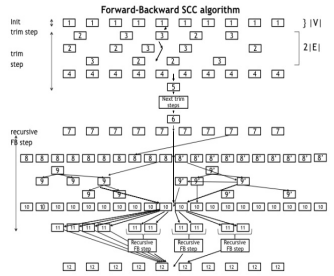
Tohoku University, Japan
Computer Science
Hiroaki Kobayashi
Akihiro Musa
Kazuhiko Komatsu
Ryusuke Egawa
Hirovuki Takizawa

Moscow State University, Russia
Research Computing Center
Vladimir Voevodin
Alexander Antonov
Dmitry Nikitenko
Alexey Teplov
Ilya Afanasyev

Background

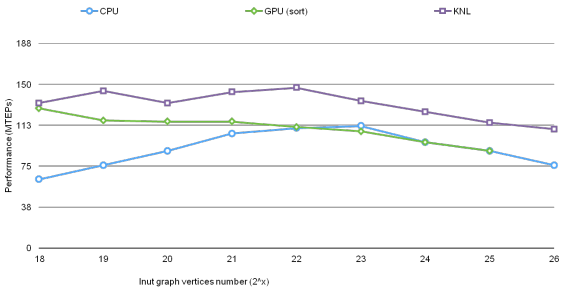
Design and development of architecture for building of exaflop-level systems is one of the most challenging tasks of contemporary high-performance computing. One of the most promising ways to find the solution of this task is using vectorization to achieve high CPU output. The origins of vectorization are found in linear memory structure of the computers and in wide usage of such data structures as arrays. Using vector operations allows much easily specifying compute blocks in terms of machine commands. The code becomes compact, and at the same time the efficiency of corresponding blocks is much higher than of those based on traditional approaches.

Implementation and adapting of many of known algorithms for vector-pipeline supercomputers, that are originally designed for vector processing, can appear more effective by a huge ratio, than same algorithm implementations for other architectures. The whole spectrum of such kind of topics is up to be studied and researched as a part of the project.



Vector data processing is widely used not only in supercomputers at present; it is widely implemented.

The figure illustrates the example of Strongly Connected Components (SCC) algorithm implementation for CPU, GPU and KNL.



The proposed by the joint team research is to be conducted at the frontier of state of the art.
✓ Japan team has been successfully undertaking designing and development of the world-best vector processing based microprocessor architectures. The team also posses all necessary experience, knowledge and technology that can grant hardware-related support for the project fulfillment.
✓ Russian team is known to possess deep and wide decades-long experience in analysis of program and algorithm structures that bases on fundamental and applied research results of a world-level, as well as rich many year experience supporting the largest in Russia HPC center, The Supercomputing Center of Moscow State University.

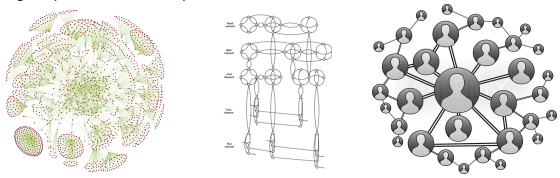
Motivation and goals

The goal of the project is the deep research on vector data processing potential that can be used both for development of new promising HPC systems and development of methods for solving of extremely large-scale numerical problems that operate huge amounts of data and intensively use memory access.

As a part of the project theory and practice of vector data processing analysis will be performed together with peculiarities of hardware implementation and software stack support. The analysis results will serve the basis for development of solving large-scale tasks methods taking in account real state-of-the art of present supercomputing which is characterized by huge performance capabilities of HPC systems, large amounts of data and extremely high and growing level of parallelism.

The developed methods will be approbated on a complex set of algorithms aimed at processing of graph structures of ultra high scale, containing over 10^9 vertices and edges (this scale can be met when analyzing social networks). These algorithms are characterized by highly intensive memory utilization, irregular memory access, absence of floating point operations, and at the same time the task scales demand precise data processing on all levels of memory hierarchy. Efficiency estimations for the methods of solving this kind of tasks on HPC systems based on vector data processing will be obtained as a part of the research results.

Graph problems are known to be the basis for solution of the wide range of applied and scientific challenges. For example: communicational and transport networks optimization, cryptography, social nets analysis, bioengineering, web data analysis and so on. These problems are very specific in sensibility of organized memory stack and special instruction sets usage to achieve good performance on required task scales.



The promising way to achieve high performance on such type of problems is to take advantage of the vector processors and machines.

Reaching project goals will provide immediate results in form of described algorithm properties such as scalability and potential of vectorization disclosure. This rich experimental data is especially valuable becoming commonly and widely available, for example, as a part of the AlgoWiki project. The combination of theoretical definitions and estimations of the most known graph algorithms and their implementations together with experimentally obtained information on their dynamical properties will provide all-round descriptions which can be used for efficient development of scientific applications that require these approaches as well as better understanding the requirements for next generation vector processors and memory architecture.

Both RU and JP teams will benefit greatly from theoretical and experimental study in the area:
✓ AlgoWiki project can be enriched with experimental data on solving graph based problems;
✓ Scalability and other algorithms properties update can contribute to the development of next generations of vector processing environment.

Early performance evaluation

To achieve all the joint project goals and results it is necessary to perform experimental runs on top level vector supercomputer.
The tables illustrate the present scales of well-known graph problems that can be solved using compute nodes of traditional heterogeneous supercomputers, and reached performance by RU team.

The diagram shows example of experimental results obtained for traditional heterogeneous nodes.

Graph Problem	MAX graph size CPU	MAX graph size GPU	MAX graph size KNL
SSSP	2^{26}	2^{24}	2^{26}
BFS	2^{27}	2^{25}	2^{26}
SCC	2^{26}	2^{25}	2^{26}
Bridges	2^{27}	2^{25}	2^{27}
MST	2^{27}	2^{27}	2^{27}
Transitive closure	2^{26}	2^{25}	2^{26}

Graph Problem	Intel Xeon CPU			NVidia GPU			Intel KNL		
	Small Size Graphs	Medium Size Graphs	Large Size Graphs	Small Size Graphs	Medium Size Graphs	Large Size Graphs	Small Size Graphs	Medium Size Graphs	Large Size Graphs
SSSP	170 MTeps	205 MTeps	121 MTeps	612 MTeps	393 MTeps	-	**462 MTeps**	**407 MTeps**	**326 MTeps**
BFS	621 MTeps	**986 MTeps**	804 MTeps	950 MTeps	718 MTeps	595 MTeps	**986 MTeps**	965 MTeps	**983 MTeps**
SCC	63 MTeps	110 MTeps	89 MTeps	128 MTeps	111 MTeps	89 MTeps	**133 MTeps**	**147 MTeps**	**115 MTeps**
Bridges	**201 MTeps**	**188 MTeps**	**134 MTeps**	139 MTeps	99 MTeps	76 MTeps	157 MTeps	131 MTeps	109 MTeps
MST	86 MTeps	85 MTeps	53 MTeps	**385 MTeps**	**309 MTeps**	**216 MTeps**	201 MTeps	188 MTeps	53 MTeps
Transitive closure	1295 MTeps	**1702 MTeps**	868 MTeps	**1711 MTeps**	1295 MTeps	**1295 MTeps**	1073 MTeps	1054 MTeps	1046 MTeps