Heterogenous Parallel Distributed Computing in Java

HPDCJ

Piotr Bała

bala@icm.edu.pl
The field of parallel and distributed computing is currently undergoing a major transformation. The largest HPC systems today contain more than 200,000 cores, and million-core systems are expected to begin arriving before the end of this decade. Only a few of the HPC site's single most important application codes scale beyond 1,000-2,000 processor cores. In the next years many HPC applications will need to be fundamentally rewritten so they can efficiently exploit HPC systems that will soon feature 1 million processor cores or more. Parallel programming using MPI or OpenMP is not easy. GPU and FPGA programming is even more difficult.
Scientific background (2)

- Numerous applications are written in Java
  - Java is very popular especially for data analysis (see poster)
- Almost all students in computer science are very early exposed to Java, much earlier than to parallel
- There is no well adopted tools for parallel programming in Java, especially for heterogeneous systems
- Java on GPUs and other multicore accelerators is still a challenge
Key challenges

- The HPDCJ project develops Java library tools and programming models for scalable parallel computing.
- Focus on the ease of use
- Make it exploitable by the huge user base of mainstream computing
- Extend the existing solution with the capability of running on the heterogeneous systems including GPU and mobile devices.
- Add fault tolerance mechanisms to the parallel Java library including fault detection and rescheduling of the application execution.
- Applicability for distributed heterogeneous systems by a set of selected, key applications including data-intensive Big Data applications.
PCJ Library

- pcj.icm.edu.pl
- Designed based on PGAS (Partitioned Global Address Space) paradigm
- Simple and easy to use
- OpenSource (available at GitHub)
- Does not introduce extensions to the Java language
  - no new compilers nor pre-processors
- Does not require additional libraries
- Good performance and scalability (beyond 6,000 cores, now 200,000 cores)
Impact

- Adoption of the solutions to the state of the art heterogeneous parallel and distributed systems will be easier and available to much larger user communities.

- Strengthen European industry and research in the supply, operation and use of heterogeneous parallel systems and will allow to achieving world-leadership.

- For new target group: students we will prepare dedicated tutorials accomplished by the webcast and on-line material.
HPDCJ consortium

- **ICM University of Warsaw** (coordinator)  
  - Poland
  - Provides PCJ library

- **IBM Research Lab Zurich**  
  - Switzerland
  - Focus on a selected applications
  - Large scale analytics with an emphasis in graph algorithms

- **Queen’s University Belfast**  
  - UK
  - School of Electronics, Electrical Engineering and Computer Science
  - Contribute new methods for scaling the JVM on many heterogeneous cores

- **Bilkent University**  
  - Turkey
  - Focuses on the stream processing capabilities
Project status - collaboration

- Project started date 1 Oct 2014
- ICM finished project Dec 2017
  - All objectives allocated to ICM fulfilled
- QUB, IBM, BU prolonged project till end of 2018
  - ICM will support partners as during the project
PCJ – memory layout

Physical node

- PCJ thread 0
- PCJ thread 1
- PCJ thread 2
- PCJ thread 3

JV
M

- a
- b[i]
- n
- X[i]
- a
- b[i]
- n
- X[i]
- a
- b[i]
- n
- X[i]

CPU
CPU
CPU
CPU

Physical node

- PCJ thread 4
- PCJ thread 5
- PCJ thread 6
- PCJ thread 7

JV
M

- a
- b[i]
- n
- X[i]
- a
- b[i]
- n
- X[i]
- a
- b[i]
- n
- X[i]

CPU
CPU
CPU
CPU

Shared variables

Local variables
PCJ – memory layout and communication

Physical node

PCJ thread 0
PCJ thread 1
PCJ thread 2
PCJ thread 3

CPU

PCJ thread 4
PCJ thread 5
PCJ thread 6
PCJ thread 7

CPU

JV
M

Shared variables

Local variables
Main scientific results

- PCJ scales up to 200,000 cores
- Graph500 implementation in Java with PCJ
- Paralelization of the Genetic Algorithm
  - Scalability up to 1500 cores
- Parallelization of the sequence alignment (PCJ-blast)
  - Scalability up to 6144 cores
- Parallelization of spare matrix multiplications
  - Scalability up to 100 cores
- Integration with JCuda
- Streaming library developed

- 16 papers published
- 1 papers in review
Dissemination

- Web page in place: pcj.icm.edu.pl/hpdcj
  - Traffic correlated to the HPDC visibility in media
  - About 4 downloads per day (1500 a year)

- HPDCJ in media
  - Presentations in highly visible media (biuletins, web pages, radio)

- Presentations to IT companies
  - Intel
  - Oracle

- Tutorials
  - Web tutorial
  - PCJ tutorials at PPAM’15, PPAM’17, SCFE’18

- Presentations for general public
  - Warsaw Science Festival, Hackathons, JUGs, etc
Dissemination

- Service for students to learn parallel programming
  - automatic verification of the code (programming contest)
  - PCJ, MPI, OpenMP
- PCJ included in the course offer for Cardinal Stefan Wyszyński University (Warsaw)
- PCJ used in summer training of University of Warsaw Computer Science students
  - parallel sort
- PCJ used in the Hackathons at ICM
- PCJ included in the Parallel Programing course at ICM
  - courses started 1st October 2016 (already 4 editions)
- PCJ included in the Parallel Programing course at NCU
  - courses started 1st October 2017
Sustainability

- The main project results are Open Source

- GitHub:  [https://github.com/hpdcj/pcj](https://github.com/hpdcj/pcj)
- Docs, manual, examples available:  [http://pcj.icm.edu](http://pcj.icm.edu)

- See our poster

*Thank you*