

Parallel Computing in Java – tutorial

Organisers and presenters:

- Piotr Bała (ICM University of Warsaw, Poland)
- Marek Nowicki (N. Copernicus University, Toruń, Poland)
- Łukasz Górski (N. Copernicus University, Toruń, Poland)
- Magdalena Ryczkowska (N. Copernicus University, Toruń, Poland)

PCJ is an award winning library for parallel computations in Java. In particular, at Supercomputing 2014, the PCJ library received HPC Challenge Class 2 Best Productivity Award, which recognize the efficient way of programming parallel applications.

The **PCJ library** (<http://pcj.icm.edu.pl>) implements partitioned global address space (PGAS) model and was inspired by languages like Co-Array Fortran, Unified Parallel C and Titanium. In contrast to listed languages, the PCJ does not extend nor modify language syntax. The PCJ library put emphasis on compliance with Java standards. The programmer does not have to use additional libraries, which are not part of the standard Java distribution. Compared to the other solutions PCJ does not need a dedicated compiler to preprocess code.

The PCJ library runs on any HPC system equipped with Java runtime environment. It shows good performance and scalability up to thousands of cores. The PCJ library has been already used to parallelize different benchmarks as well as large scale parallel applications such as raytracing, FFT, MapReduce or Genetic Algorithm.

Writing PCJ application is simple. It can be built in the form of a single class which extends **Storage class** and implements **StartPoint** interface. The **Storage class** can be used to define shared variables. **StartPoint** interface provides necessary functionality to start required threads, enumerate them and performs initial synchronization of tasks. PCJ provides methods to synchronize tasks, one-sided asynchronous data transfer between tasks (**put** and **get** methods) as well as broadcast.

The presenters running tutorial have long standing research expertise in parallel and distributed computing including traditional HPC background. They are running regular courses for the students as well as training sessions for the HPC users.

Schedule:

The 4 - hour tutorial comprises three parts parts, with time for discussion at the included coffee breaks. The tutorial is organized in the form of hands-out with the short introduction. We would like to get participants involved in the programming of simple examples provided by the organizers as well as computational kernels brought by the participants.

- Part 1 (1 hour) briefly covers the PGAS programming model and presents details of the PCJ library. It provides overview of the library and presents examples of the typical programming tasks. Examples of the parallelization of the most popular programming kernels will be provided.
- Part 2 (1 hour) is design as practical introduction to the PCJ. Participants will write and execute parallel applications starting from the “Hello world”, trough different communications schemas up to the parallelization of the selected computational kernels.
- Part 3 (2 hours) will focus on more advanced applications including problems suggested by the participants.

Requirements

The workshop will be run on the standard PC computers with the access to the Internet. Java SDK 1.8 (or 1.7) is required as well as Java IDE (Netbeans are suggested, Eclipse and IntelliJ are an option). The participants will have possibility to submit jobs to the large HPC systems using on-line tools developed at ICM to support education.

The participants can use their own computers with the Java SDK 1.8 and Netbeans (or another IDE) installed. They should have possibility to connect to the internet.

We would like to ask participants to provide (if possible) information on the computational problems they would like to parallelize.