FastFlow is a parallel programming framework for multi and many core platforms based upon non-blocking synchronization mechanisms. The framework is structured as a stack of layers that provide different levels of abstraction to the application programmer. FastFlow provides the parallel programmer with a set of ready-to-use, parametric algorithmic skeletons modeling the most common parallelism exploitation patterns.

**Flexible Layered Design**

FastFlow is a C++ class library designed as a stack of three main layers. The different layers have two main purposes: 1) promoting high-level parallel programming, i.e. explicit pattern-based parallel design of applications, and 2) to be flexible and efficient for programming multi and many core platforms. The FastFlow programmer may choose to use the mechanisms provided by the layers which best suit his/her needs.

Building Blocks: This is the lowest level layer. It comprises: i) the wrapper nodes (the ff_node derived classes) used to embed existing portions of code (C/C++, OpenCL, CUDA) into parallel programs; and ii) the one-to-many, many-to-one and feedback combinators for connecting nodes and routing data in different ways. At this level any asynchronous streaming network can be built; the semantics is data-flow. Each wrapper building block has an input and an output lock-free bounded or unbounded SPSC queue, and their well-defined semantics promotes the possibility to automatically refactor their compositions to better exploit target architecture features.

Core Parallel Patterns: On top of the Building Block layer, has been implemented the pipeline and several forms of the task-farm implementation skeletons. They can be composed and nested in several different ways using also the feedback component for routing back data streams. The task-farm is fully customizable both in terms of scheduling and gathering policies. A pipeline of task-farm components, when nested, may be optimized for reducing the number of concurrent threads.

High-Level Parallel Patterns: This is the top layer level. To structure his/her parallel application, the application programmer, uses the patterns available at this level and their compositions through the pipeline and task-farm core patterns. This layer is still in evolution: more patterns will be developed and further optimizations will be applied to the current ones. The high-level patterns currently available are ParallelFor/Map, ParallelReduce, Stencil, ParallelSearch, Macro-Datadflow, D&C, Pool Evolution.

**Platforms supported**

FastFlow supports Intel, AMD, IBM Power and ARM general purpose multi-core based platforms. Recently it has also been ported on Tilera TilePro64 and Intel Xeon PHI such that tasks can be offloaded on these accelerators. FastFlow-based code may be compiled with any recent GNU and Intel compilers.

**FP7 Projects using FastFlow**

- REPARA

**References**

Project Home:  
→ http://mc-fastflow.sourceforge.net  
→ http://calvados.di.unipi.it/fastflow

SVN repository:  
→ https://svn.code.sf.net/p/mc-fastflow/code