FastFlow: multicore and GPGPU programming made easy

University of Torino & University of Pisa, Italy * http://sourceforge.net/projects/mc-fastflow * Open source software under LGPLv3

Motivation

The whole computer hardware industry embraced parallel platforms, such as multicore, GPGPUs, and cloud. For these platform, the extreme optimisation of sequential algorithms is no longer enough to squeeze the real machine power. In the long term writing parallel programs ought to be as efficient, portable, and correct as it has been to write programs for sequential computers. To date, however, the parallel programming drill does not embrace much more than lowlevel communication libraries. In the hierarchy of abstractions, it is only slightly above toggling absolute binary into the front panel of the machine. By definition, the raison d'être for parallel computing is high performance, but speed-up need not be the only measure. Human productivity, total cost and time to solution are equally, if not more important.

Material and methods

FastFlow is a C++ parallel programming framework advocating high-level, pattern-based

parallel programming. It chiefly supports streaming and data parallelism, targeting heterogenous platforms composed of clusters of shared-memory platforms, possibly equipped with computing accelerators. The FastFlow run-time support efficiently support fine grain parallelism via non-blocking multi-threading with lock-less synchronisations; zero-copy network messaging; asynchronous GPGPU offloading.

FastFlow team

M. Aldinucci, M. Torquati, M. Danelutto M. Drocco, C. Misale, G. Peretti Pezzi, F. Tordini



Signal filtering

A high-performance filter for realtime video denoising. The filter is composed of two phases: detection (e.g. classical median filter) and correction. The filter achieve very good restoration quality, comparable to jpeg compression. Unlike traditional methods, the correction phase is based on a variational method and succeeds to restore also extremely noisy images (up to 90% of noisy pixels). FastFlow provided seamless portability to CUDA boards (e.g. NVidia K-40). This work has been presented at Nvidia GTC 2014.



High-level pattern-based parallel programming for C/C++

Data parallelism: ParallelFor, MapReduce, StencilReduce

Data parallel patterns can be easily deployed onto multi-cores and multiple GPGPUs. No additional programming effort with respect to OpenMP.

// FastFlow (--std=c++11) ff::ParallelFor pf; pf.parallel_for(0L,N,[&A](const long i) { A[i]+=1; ,nworkers);

A[i]+=1;

Stream parallelism: Pipeline, Farm

Stream parallel patterns can be deployed onto multi-cores, distributed platforms, and clouds (e.g. Amazon EC2). Stream patterns can be composed to model arbitrary streaming networks, and can be nested with data parallel patterns. This makes it possible to use the aggregate power of (physical or virtualised) heterogenous clusters of multicore and GPGPUs. FPGA support is currently experimental.

Platforms and OSes support

Being a plain header-only C++ library, FastFlow can be used (or easily ported) on any homogenous or heterogeneous parallel platform. Among the others, the following platforms are supported:

Intel x86_64 - Linux/MacOS - gcc/clang/icc Intel x86_64 - Windows 7/8 - MS Visual Studio Arm - Linux - gcc Arm - iOS - clang (experimental) IBM Power - Linux - gcc Accelerators: NVidia/CUDA/OpenCL, Intel Phi, Tilera Tile64

DNA alignment (Bowtie2 & BWA)

Bowtie2 and BWA are among the fastest and most used alignment tools for genome analysis in bioinformatics. In the FastFlow porting, the concurrency structure has been redesigned passing from a pool of thread accessing shared data to a Farm/Master-Worker equipped with automatic memory affinity scheduling. The Fastflow port differs from original code no more than a dozen of code lines but gains up to ten speedup points over the original parallel, hand-optimised software.



Hi-C data analysis has emerged as a powerful technique to understand how the genome is packaged in cells to control gene expression. nuChart-II provides a gene-centric view of the 3D chromosomal neighbourhood. Starting from the sequential C++ implementation, the graph exploration loop has been parallelised by using the ParalleFor pattern provided by FastFlow, that permitted to improve the performance with minimum effort.



// OpenMP (-fopenmp) #pragma omp parallel for num_threads(nworkers) for(long i=0;i<N;++i) {</pre>



Next-gen DNA analysis (nuChart-II)





Performance

Like other high-level programming frameworks, such as Intel TBB, OpenMP, Hadoop, FastFlow simplifies the design and engineering of portable parallel applications. However, it has a clear edge in terms of expressiveness and performance with respect to other parallel programming frameworks in specific application scenarios, including, inter alia:

- and multi-core
- algorithms







Projects & partnerships, over 10M€ 22,529 2.02 81,457 Bounce Rate Avg. Visit Duration 00:02:26 66.20% HPC HIPEAG **HPC500** REPARA Compagnia di San Paolo SEVENTH FRAMEWORK