Raising programming level abstractions to target modern parallel architectures

M. DANELUTTO, P. DAZZI,
G. MENCAGLI, M. TORQUATI,
A. OTTIMO, N. TONCI

DEPT OF COMPUTER SCIENCE – UNIV OF PISA

HPC KEY TECHNOLOGIES AND TOOLS LAB EVENT SEPTEMBER 14, 2022

Structured parallel programming

- Raise the level of abstraction presented to parallel application programmers
- Move implementation details and complete hardware targeting in tools (compiler + runtime)

- More abstract and declarative programming style conveys much more information to tools that can be exploited to orchestrate and integrate know solutions
- Same as happened in the last century for sequential programming

Our experience
(with previous group members)

Mostly theoretical: last decade last century

- Algorithmic skeletons, P3L, Muskel, template based and macro data flow implementations

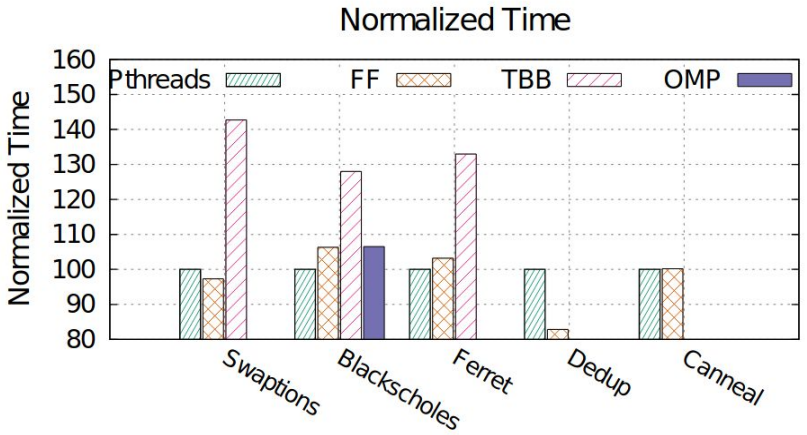
First Projects: beginning current century

- ASSIST

Going mainstream: from '10s

- FastFlow, Accelerator targeting, Optimization tools, DSLs (WindFlow), Autonomic management (Nornir)

FastFlow in a nutshell



- *Structured*: parallelism expressed through pattern composition
- *High level*: patterns modelling common, useful and efficient parallel patterns
- *Layered*: high level patterns (application programmers), building blocks (system programmers, DSL designers)

- *Multitarget*: shared memory multicores, accelerators (GPU & FPGA), COW/NOW
- *Host language*: "modern" (17 and up) C++
- *Header only:* include and compile –O3
- *Performance*: comparable with state of the art, lower level frameworks

The FastFlow ecosystem

- Nornir: farm with autonomic, best effort enforcement of user provided power/performance (XML) contract (DeSensi, now at La Sapienza)
- WindFlow: data stream parallel DSL framework, with low latency and high throughput
- RplSh: high level refactoring fo pattern expressions with different non-functional property evaluation
- Distributed FastFlow: structured orchestration of structured patterns on top of MPI or TCP/IP
- FPGA targeting: patterns to compose, orchestrate and optimize kernel design and implementation
- SPAR: annotated c++ targeting FastFlow
- Projects: ParaPhrase (FP6), REPARA (FP7), RePhrase (H2020) (exclusive), Admire, Textarossa, EUPEX (contribute) ...

Advanced research lines

Edge / Continuum optimizations

- Parallel structure & requirements/contracts exposed to tools and run time config
  - Computation vs data displacement
  - Security and privacy enforcement
  - Distributed heterogeneous orchestration

Accelerator management

- Parallel structure drives
  - Kernel compilation and deployment
  - Kernel copy management
  - Kernel composition
  - Memory buffer and copy management

Advanced research lines (2)

High level tools

- Formal reasoning on pattern composition
- Targeting different non-functional properties
- Design *ab initio* good solutions
- Solution space exploration before coding

DSL & libraries

- AI specific patterns (composition of parallel building blocks)
- Specific patterns targeting heterogeous, distributed edge to cloud target architectures (hierarchical/à la two tier rule)