ITEA 2

INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

ParCo2009

ParMA

Parallel Programming for Multi-core Architectures

www.parma-itea2.org

The need for parallel computing

- **Computational power is instrumental for:**
  - European **Research** (life science, climate, …)
    to attack some of the most challenging problems in basic sciences
  - European **Industry** (avionics, energy, automotive, etc.)
    to reduce time-to-market & prototyping costs while improving quality
    by advanced simulation and modelling
- **Every year we get ~~faster~~ more processors**
  - The multiplication of cores (processors) in new computers
    forces developers to write parallel programs
  - Concurrency is complicated and programmers are not well equipped to parallelize and optimize applications
- So, the **ParMA** Project aims at:

Developing and validating a technology
that enables compute-intensive applications
to take full advantage of the power of multi-core architectures

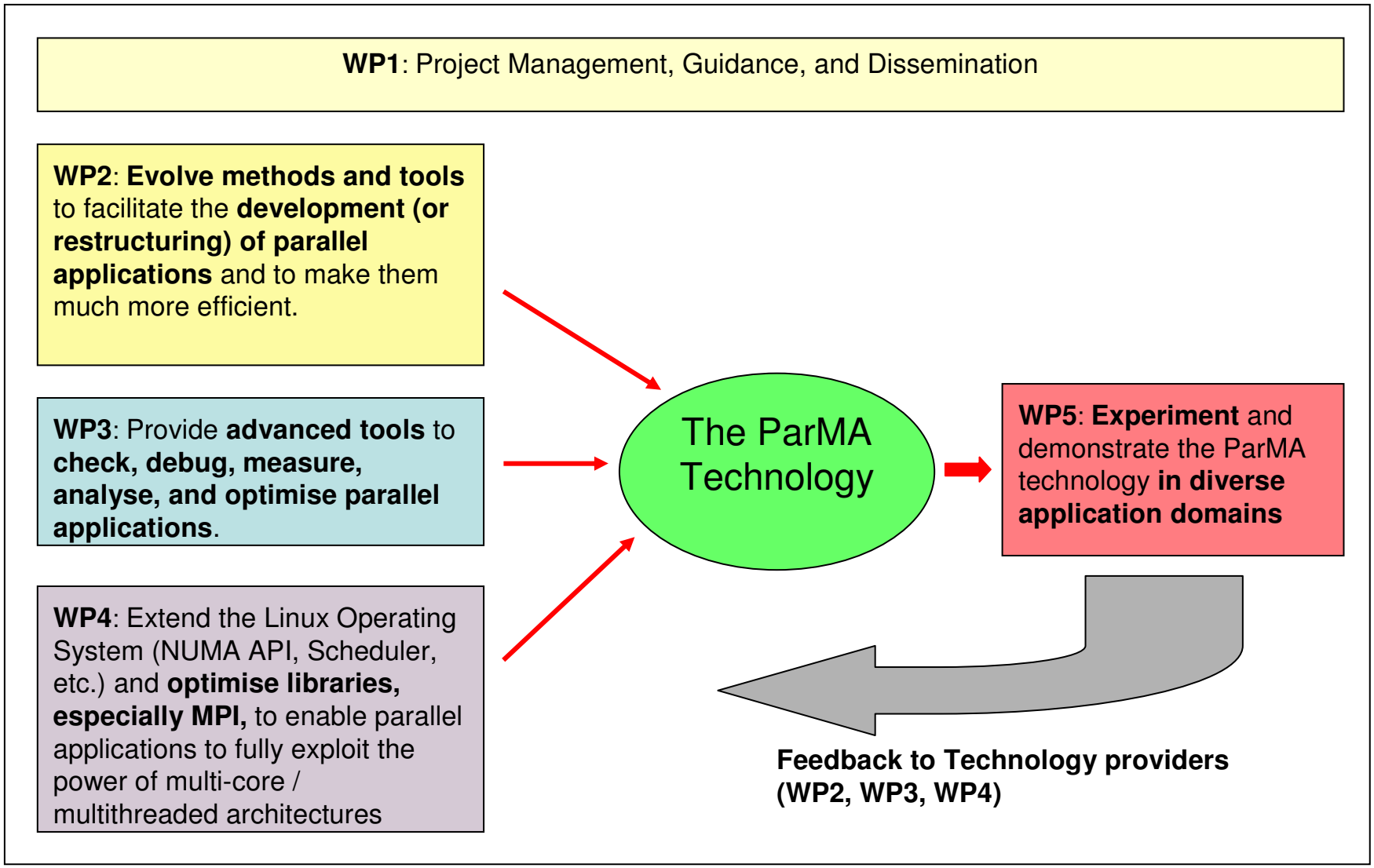
The issues to be solved

To meet new scientific and engineering challenges, we mainly need:

- faster, more accurate computational mathematics
- new parallel programming models,
- methods and tools to detect and extract parallelism (thread extraction),
- enriched directives to express parallelism,
- easy to use and scalable performance-analysis tools,
- powerful correctness checker and debugger,
- optimized libraries for multithread / multi-core, and optimization tools
- enhanced thread management, job scheduling, …

=> ParMA is addressing all of these issues

The ParMA approach



ParMA Objectives (1 / 4)

Evolve design and programming models for parallel applications

- **Multi-level parallel programming & execution model for embedded applications by CEA-LIST / LaSTRE**
- **Hybrid parallel programming: convert OpenMP programs to hybrid programs (OpenMP + MPI) by IT-SudParis**
- **Application restructuring and code optimization by CAPS and UVSQ**
- Adaptation of message passing APIs to MPSoC environments by **UAB**
- Method & Tool to design interconnection Networks on Chip (NoC) by **UAB** & **Robotiker**

ParMA Objectives (2 / 4)

Provide parallel programming tools that apply to any application domain and any parallel architecture, and integrate them

- **MPI Correctness checker: MARMOT from HLRS, to support full MPI-2 Std, parallel IO, one-sided com, C++**
- **Tracing facilities: Open Trace Format;** Peruse (MPI extension)
- **Performance-analysis toolset for large scale applications**
  - **KOJAK / SCALASCA from JSC**
  - **Vampir from TU Dresden (ZIH) & GWT**
- **DDT** (Distributed Debugging Tool) to deal with large number of threads and OpenMP and **OPT** - Optimization & Profiling Tool, from **Allinea**

ParMA Objectives (3 / 4)

Push operating systems to support new multi-core architectures

- Linux extensions by **Bull:**
  - Improve shared and cache memory management to minimize buffer copy and optimize bandwidth
  - Improving cache coherency for better latency
  - Monitor timer and synchronise daemons to reduce disturbar
  - Optimise parallel I/O performance with Lustre
- Setup and manage a common HPC platform (**Bull**)
- Optimization of linear algebra functions by UVSQ
- Build an MPSoC embedded platform

ParMA Objectives (4 / 4)

Demonstrate the technology in various application domains

- Metal forming simulation with INDEED by
- Casting process simulation with MAGMASOFT by
- **3D-combustion simulation by RECOM-AIOLOS by**
- HPC benchmark by
- **Avionics: - Tracking algorithm**
  - Controlled simulation of systems) by
- Software Defined Radio by
- Virtual reality for manufacturing processes
  (optimisation of Collision Detection & Dosimetry) by

Main results achieved so far

- We specified and implemented a set of advanced parallel programming **models**, **methods**, and **tools.**
- In particular: the performance analysis tools, that were separated, have been:
  - Extended to support MPI-2.1 and to run in more environments
  - Enhanced and made easier to use,
  - Integrated to facilitate combined overall & detailed analysis
- Application developers adopted new algorithms, restructured, optimized their code with the help of our tools, and got significant performance improvements, up to 4x.

Conclusion

- The next presentations will give you more details on the ParMA achievements:
  - Methodology for Application Performance Tuning (Bettina Krammer, UVSQ)
  - How to Accelerate an Application: a Practical Case Study in Combustion Modelling (Benedetto Risio, RECOM)
  - From OpenMP to MPI: first experiments of the STEP source-to-source transformation (Daniel Millot, IT-SudParis)
  - Using Multi-Core Architectures to Execute High Performance-Oriented Real-Time Applications (Christophe Aussagues, CEA-LIST)
  - An Interface for Integrated MPI Correctness Checking (Tobias Hilbrich, TUD-GWT)
  - Enhanced Performance Analysis of Multi-core Applications with an Integrated Tool-chain (Thomas William, GWT)

Thank you for attending this mini-symposium