PEPPHER
Programmability & Portability

HiPEAC COMPILATION ARCHITECTURE

EPoPPEA @ HiPEAC 2012
January 24, 2012
Paris, France

PEPPHER: Performance Portability and Programmability for Heterogeneous Many-core Architectures

Sabri Pllana
(project coordinator)
University of Vienna

This project is part of the portfolio of the
G.3 - Embedded Systems and Control Unit
Information Society and Media Directorate-General
European Commission

www.peppher.eu

Contract Number: 248481
Total Cost [€]: 3.44 million
Starting Date: 2010-01-01
Duration: 36 months

SEVENTH FRAMEWORK PROGRAMME

Project Consortium

PEPPHER

- Universities
  - University of Vienna (coord.), Austria
  - Chalmers University, Sweden
  - Karlsruhe Institute of Technology, Germany
  - Linköping University, Sweden
  - Vienna University of Technology, Austria
- Research center
  - INRIA, France
- Companies
  - Intel, Germany
  - Codeplay Software Ltd., UK
  - Movidius Ltd. Ireland

Addressed Issue(s)
and
the PEPPHER Approach

Context and Motivation

PEPPHER

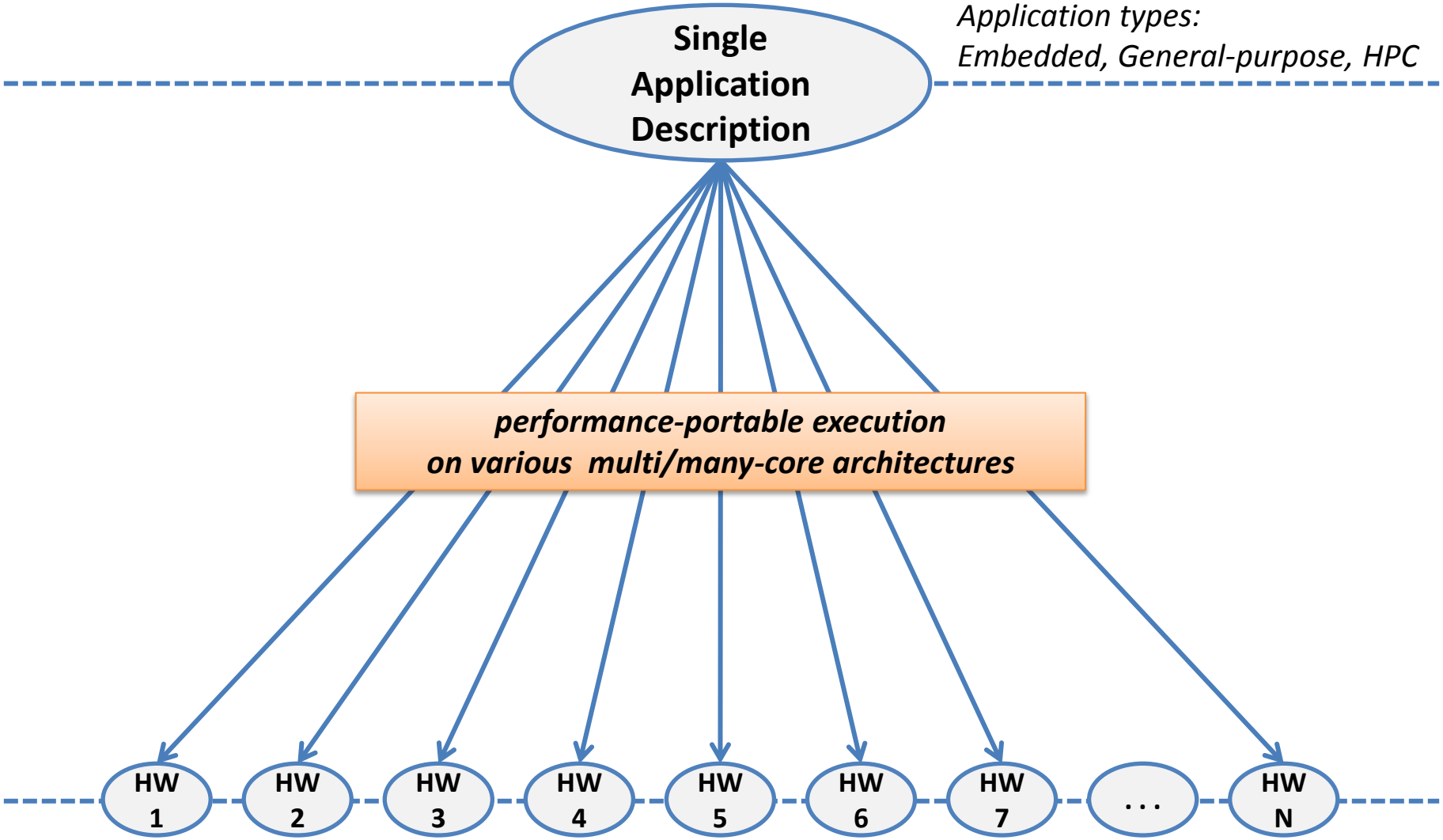
- PEPPHER addresses heterogeneous systems
  - Singe-node (instance: CPU and GPU/MIC*)
  - Single-chip (instance: APU, Cell BE)
- We do not propose a new programming model/language
  - different programming models/languages may be suitable for different core types
- Aim: enable combination of existing programming models/languages

CORA Node
Two quad-core CPUs and three GPUs (2x C2050 and 1x C1060)
Research Group of Scientific Computing, University of Vienna

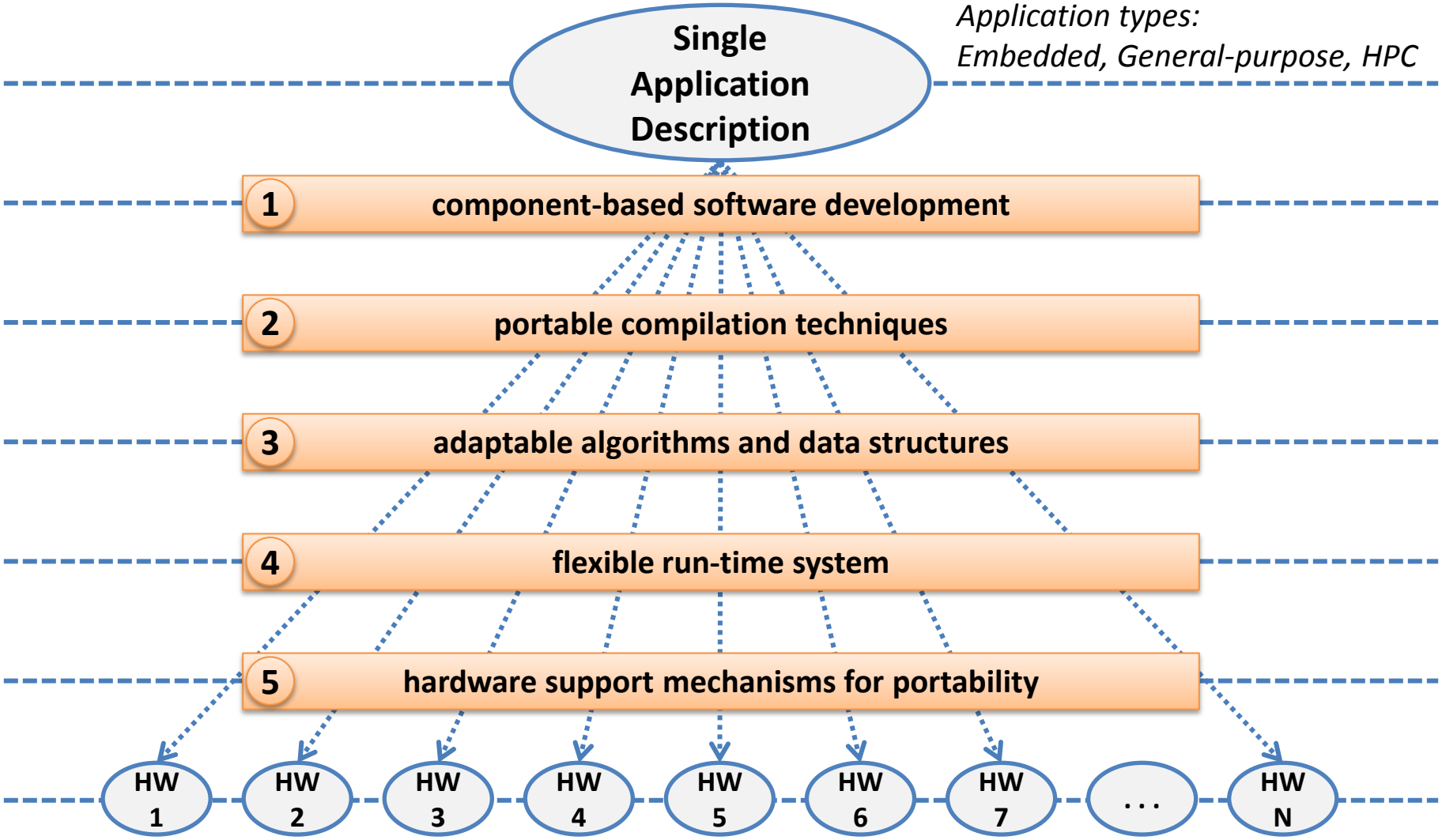
* Intel® Many Integrated Core Architecture. Intel is a trademark of Intel Corporation in the U.S. and/or other countries. (www.intel.com)

Aim: Application Runs on Any Hardware

PEPPHER



PEPPHER Holistic Approach: Five Layers



Overview

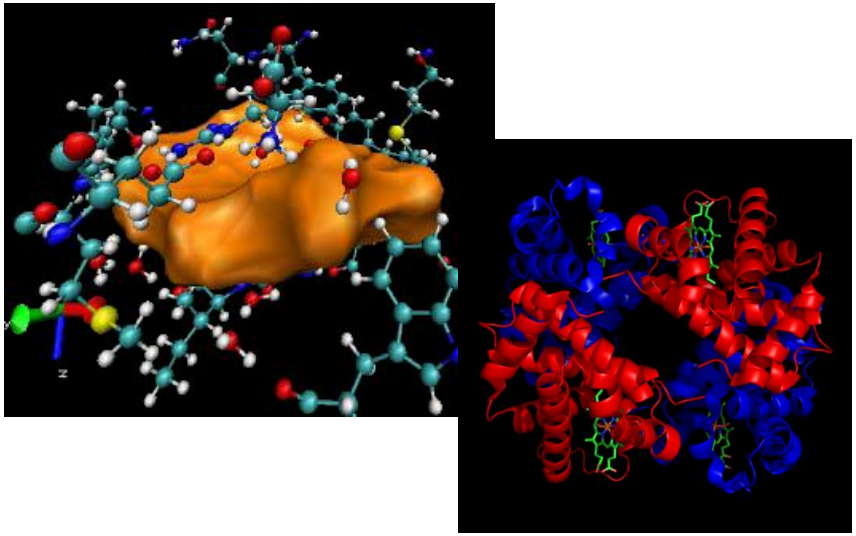
Applications

PEPPHER

Applications

Embedded, General Purpose, HPC

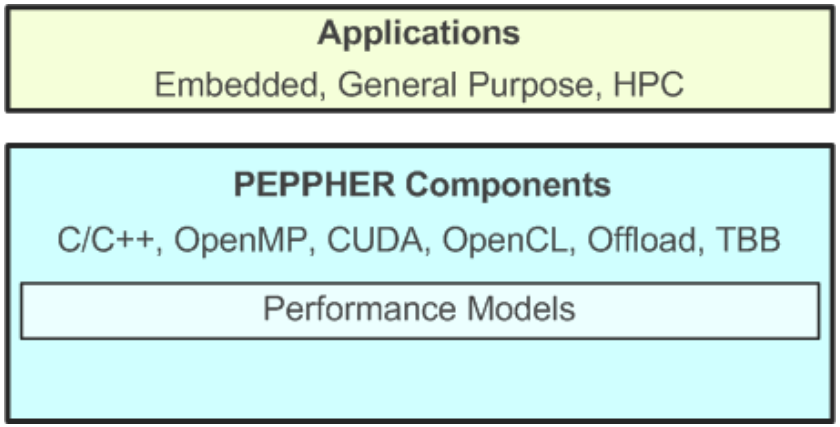
Software optics



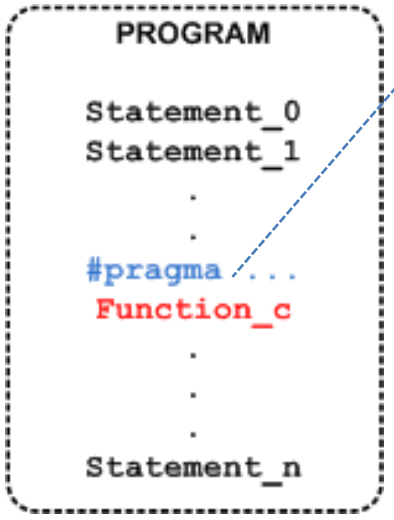
Molecular dynamics simulation

- PEPPHER targets applications from various domains
  - from small kernels to larger programs
- Applications
  - KIT: Suffix array construction
  - UNIVIE: Bzip2, OpenCV
  - Codeplay: Bullet (games physics simulation)
  - Movidius: Computational photography
  - Intel: GROMACS
- Kernels
  - INRIA: FFT
  - INRIA: MAGMA/PLASMA (QR)
  - INRIA: RODINIA (CFD solver)
  - KIT: STL (sort, find, random_shuffle)

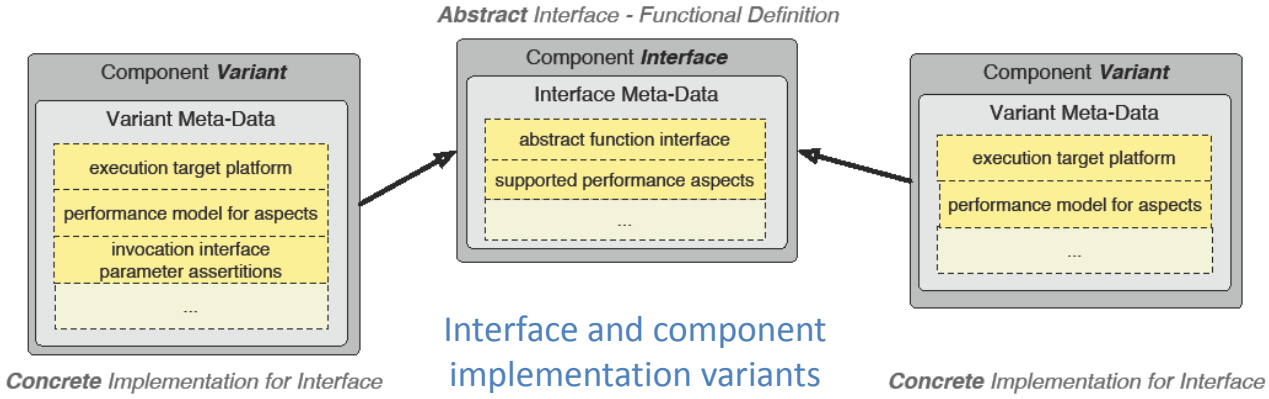
1: Component-based SW Development



- PEPPHER component
  - implements a specific functionality declared in an interface
  - annotated software module
  - rich meta-data (e.g. perf. model)
  - performance-aware
- Implementation variants
  - target execution platform
  - algorithm
  - programming model/language
  - …

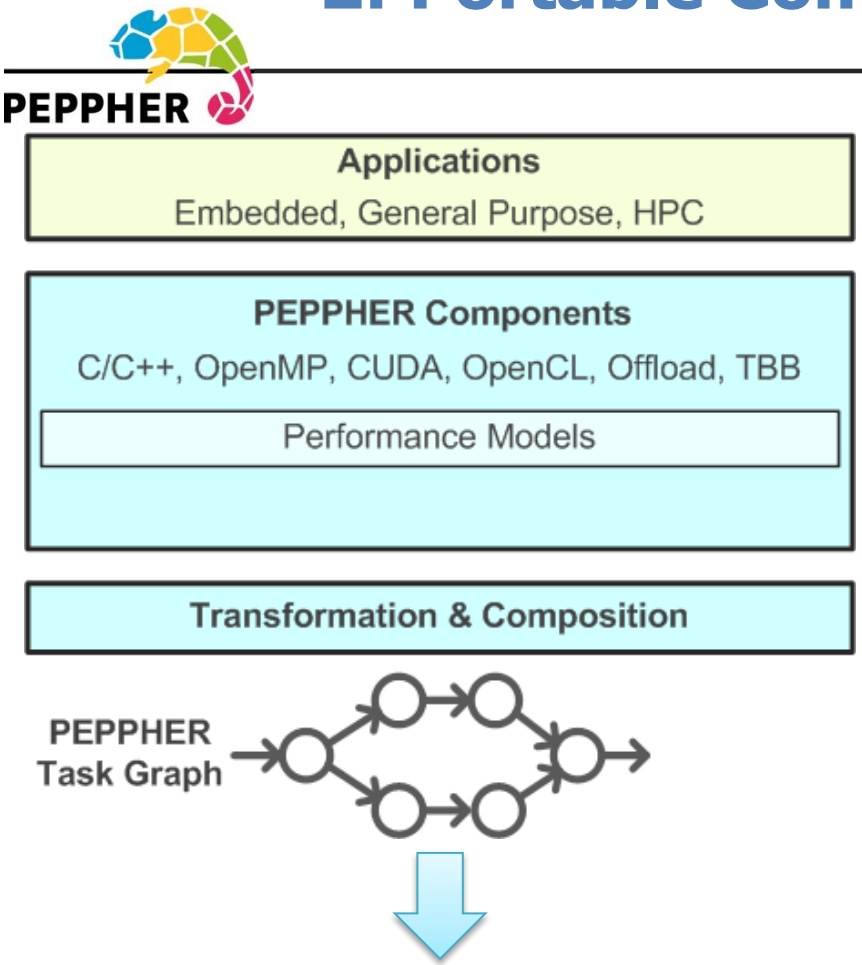


Performance-critical parts of program ("hot spots") are candidates for PEPPHER components



Interface and component implementation variants

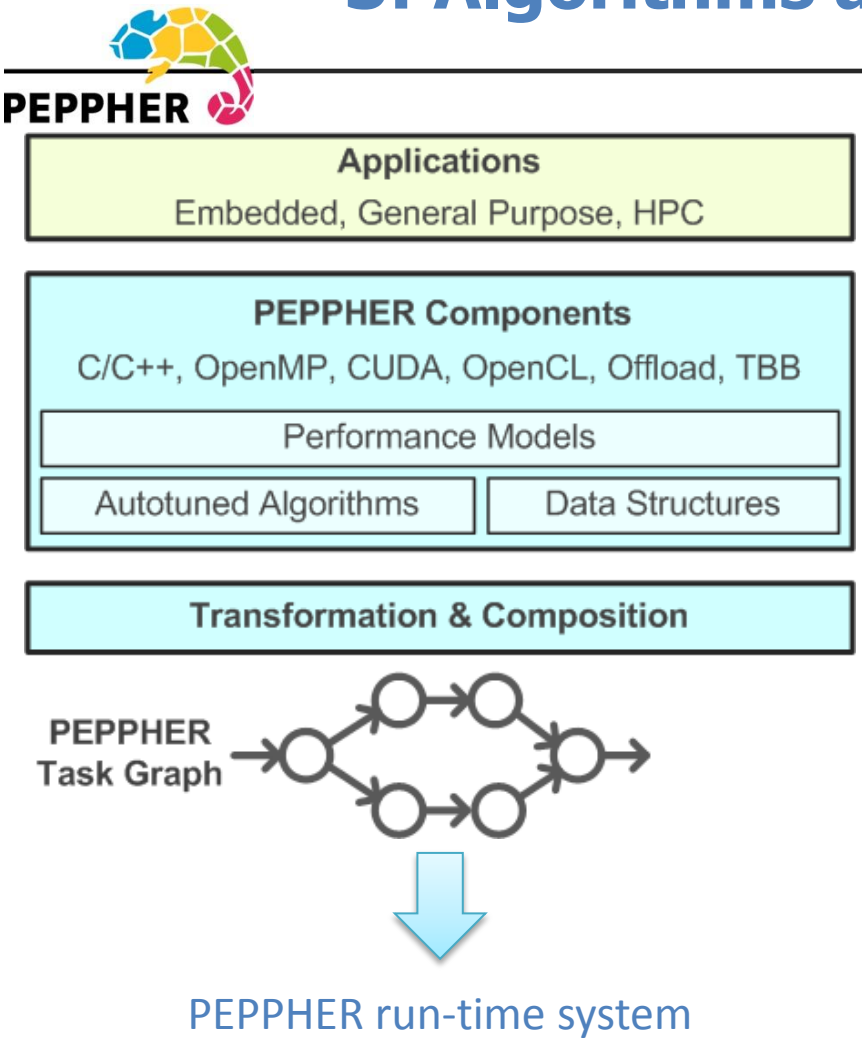
2: Portable Compilation Techniques



- Source to source transformation
  - target the run-time system
  - generate/preselect component implementation variants
- Offload C++
  - compiler and run-time system for offloading parts of C++ applications to run on accelerator cores
- OffloadCL
  - generates OpenCL code for host and OpenCL device from annotated Offload C++ source code

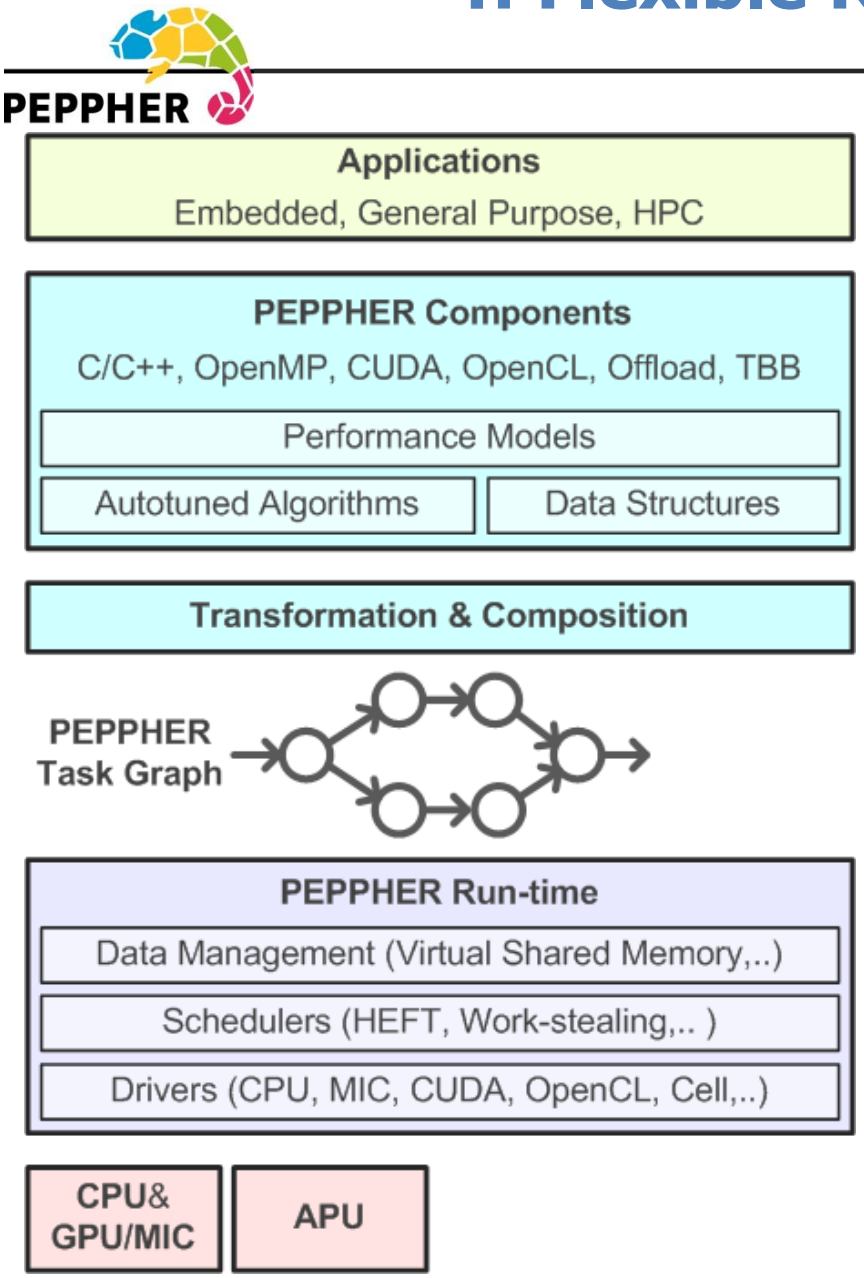
Offload™ is a trademark of Codeplay Software Ltd (www.codeplay.com)

3: Algorithms and Data Structures



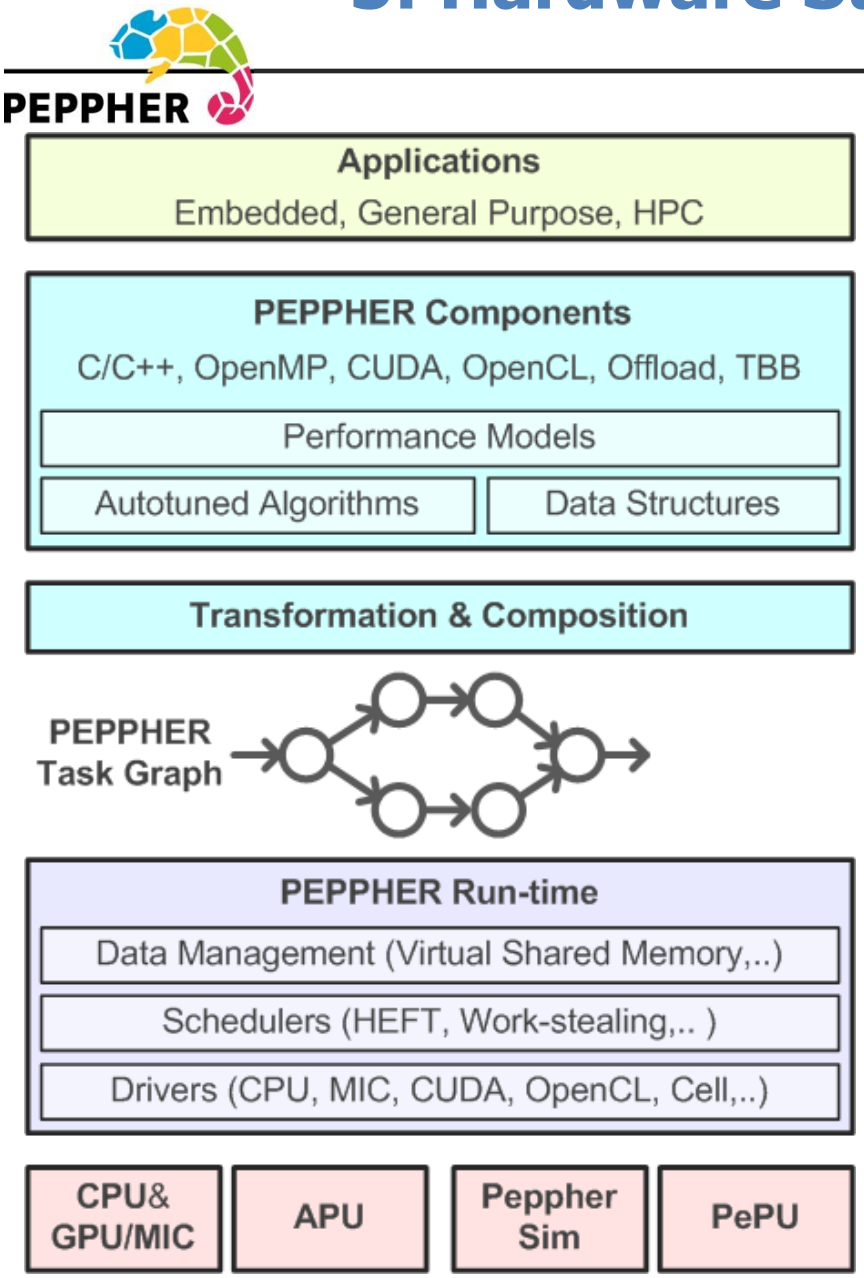
- Adaptable algorithms and data structures
  - expert-written libraries
  - provide component implementation variants
- Algorithmic toolbox
  - generate different implementation variants based on compile-time architecture-dependent tuning parameters
- Synchronization library
  - lock-free templated data structures for CPU and GPU

4: Flexible Run-time System



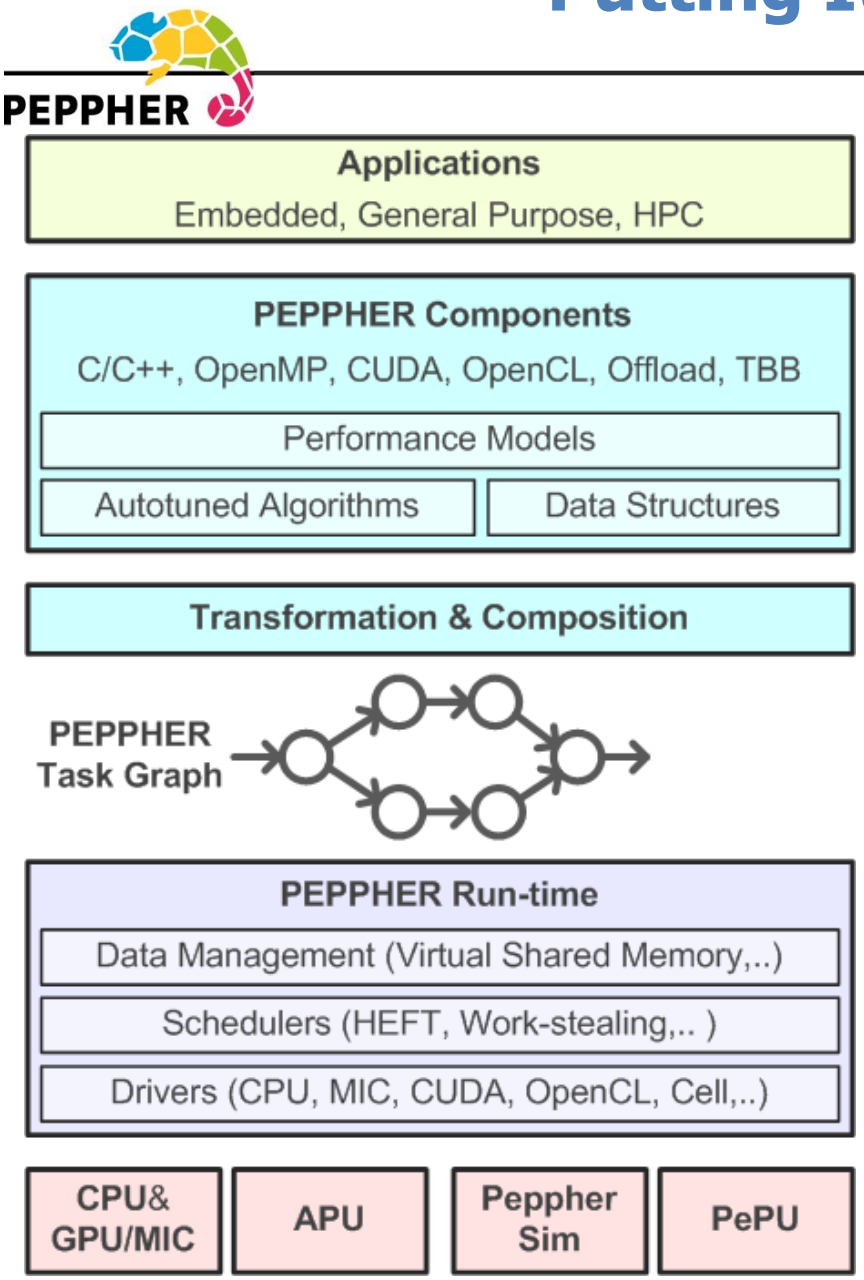
- PEPPHER run-time system
  - component meta-data supports scheduling decisions
  - dynamic scheduling of tasks on a pool of heterogeneous cores
  - provides a Virtual Shared Memory subsystem
  - provides performance feedback
- Open scheduling platform
  - scheduling algorithm = plug-in
- Tasks
  - data input & output
  - dependencies with other tasks
  - multiple implementations (GPU, CPU)

5: Hardware Support Mechanisms



- PeppherSim
  - simulates existing or conceptual architectures
  - provides an OpenCL interface
  - supports temporal and energy metrics
  - enables investigation of new synchronization primitives
- Integration of run-time system with PeppherSim
  - generation of temporal and energy-consumption performance models
- PePU (work in progress)
  - PEPPHER Processing Unit
  - a demonstration hardware platform
  - comprises multiple Movidius SABRE SoCs with an FPGA

Putting It All Together



- Applications: C/C++ source code with annotated components
- Component implementation variants for various hardware, input characteristics, and optimization criteria
- Variants may be parallelized in the most suitable framework, or supplied by "expert" programmers as part of libraries
- Transformation and compilation techniques support the variant generation/preselection
- Intermediate representation: component task graph with explicit data dependencies
- PEPPHER run-time selects dynamically variants and schedules on the available resources
- Hardware mechanisms for synchronization and performance monitoring

Expected Impact and Beyond PEPPHER

Expected Impact

- Strengthen the European excellence in heterogeneous multi-core systems
  - high-level software development, compilation technologies, algorithms and data structures, run-time systems, hardware support for programmability and portability
- Industrial use of results
  - OffloadCL by Codeplay
  - PeppherSim and PePU by Movidius
  - potential take-up of interesting PEPPHER technology by Intel
- Academic use of results for research-driven teaching
  - deliver state-of-the-art knowledge from this domain to students

First achievements

✓ PEPPHER source-code transformation system

✓ Tuned sorting algorithms for multi-core and GPU with world-leading performance

✓ OffloadCL compiler generates OpenCL code from annotated Offload C++ source code

✓ StarPU-based run-time system supports various schedulers, target devices, power-based optimization

✓ PeppherSim simulator supports temporal and energy metrics

Beyond PEPPHER

- Address fundamental parallel programming issues
  - funding scheme: FET Open, ERC grants,..?
- Investigate resource-aware parallel programming techniques
  - energy-awareness
  - architectural support for resource-efficient parallel programming
- Develop intelligent software development environments
  - programming environment supports proactively the programmer
  - automation & autonomy
  - Pllana et al. LNCS 5415, pp. 137–147, Springer 2009

Acknowledgments

- European Commission (ec.europa.eu)
- HiPEAC Network of Excellence (www.hipeac.net)
- PEPPHER Consortium (www.peppher.eu)

Some of the consortium members (from left): D. Moloney, E. Marth, S. Pllana, V. Osipov, M. Wimmer, B. Bachmayer, P. Tsigas, J.L. Träff, C. Kessler, J. Singler, S. Benkner, D. Cederman, U. Dastgeer, H. Cornelius, S. Thibault, A. Richards, M. Sandrieser, U. Dolinsky, R. Namyst, C. Augonnet, H.C. Hoppe