PEPPHER Composition Tool
Performance-Aware Dynamic Composition of Applications for GPU-based Systems

Usman Dastgeer  
Lu Li  
Christoph Kessler

Linköping University, Sweden

GPU-based systems

Challenges
- Portability  
- Programmability  
- Performance portability

PEPPHER approach
• Flexible and extensible component model for annotation of performance-critical user code (also parallel / platform-specific)  
• Autotunable algorithms  
• Runtime system (StarPU)  
  - Dynamic implement, selection, resource allocation, scheduling

“PEPPHERing” an Application (1 step):

PEPPHER Component Model
• External annotation of code in XML descriptors  
  - Interface descriptor  
  - Implementation descriptor (one per implementation variant)  
• Interface metadata  
  - Function parameter types and access mode  
  - Context descriptors, training data generators for off-line tuning  
• Implementation variant metadata  
  - Various C/C++ based programming models supported  
    - sequential, OpenMP, CUDA, OpenCL, ...  
  - Compilation / deployment information  
  - Provided and required interfaces  
  - Platform model and resource requirements  
  - Tunable parameters  
• Binding points: Component invocations (on CPU only)  
  - For component operands: C/C++ native data types or smart containers: Vector, Matrix, ...

PEPPHER Composition Tool
• Parsing+representation of component descriptors  
• Internal optimizations and adaptive offline tuning  
• Generation of off-line measuring and tuning code  
• Generation of stubs for implementation selection interfacing to PEPPHER runtime system  
• Overall coordination of the build process

Adaptive Off-line Tuning
• Learn selection (dispatch) function  
  - At component deployment time  
  - Customized adaptive decision tree  
  - Reduce measurement and dispatch overhead

Selected publications

Further information: www.peppher.eu