A Highly Efficient Adaptive multi-Processor Framework

General Project Presentation

Andreas Brokalakis
Synelixis Solutions Ltd

FPL Workshop
Chania, Crete, 7 September 2011
Presentation Outline

- What is HEAP
- Objectives
- Consortium
- Budget/Funding
- Technical Approach
- Work Structure
- First Year Achievements
- Dissemination & Exploitation plans
What is HEAP?

- Parallel code has always been considered difficult to develop
  - Shift towards multi-core architectures
  - Increase performance by focusing on throughput rather than single-thread acceleration

- Huge performance and power improvements can be achieved by tailoring the memory hierarchy, in particular the cache architecture, to the application at hand.

- The HEAP project faces these challenges directly, by providing:
  - An innovative toolset that helps software developers profile and parallelize an existing sequential implementation, by exploiting top-level pipeline-style parallelism.
  - A highly configurable cache architecture that can be tailored to an application by using the same, or similar, profiling data as those that were used for parallelization, in order to fully exploit the available computing power.
HEAP Objectives

• To speed up the programmer time required to obtain a parallel implementation of a sequential application, while achieving the same level of performance of a fully manual procedure.

• To virtually eliminate the amount of architectural expertise required to identify the best cache architecture to support the parallelized code.

• To provide a complete framework for implementing multi-core systems in different application domains with optimal cache/synchronizing sub-systems that can also be altered in real-time.

• To provide a complete open-source and commercial development and verification framework for parallel systems.
HEAP Consortium

- ST Microelectronics (STM)  IT  Industry
- Synelixis Solutions (SYN)  GR  SME
- Thales Communications (Thales)  FR  Industry
- ACE (ACE)  NL  SME
- Compaan Design (Compaan)  NL  SME
- Politecnico Di Torino (Polito)  IT  University
- ATHENA Industrial Systems Institute (ATHENA)  GR  Research Center
- Universita Degli Studi Di Genova (UniGe)  IT  University
- Singular Logic (SiLo)  GR  Industry
- Uppsala Universitet  SE  University

Well-balanced consortium:
3 Large Industries, 3 SMEs, 1 Research Center, 3 Universities
HEAP Consortium: ST Microelectronics

• Relevant Expertise
  – Design and implementation of numerous parallel systems incorporated in both embedded systems and high-end parallel frameworks and the associated programming toolsets

• Main Project Role
  – Project Administration
  – Provide requirements for parallel systems and demonstration applications
  – Evaluate and exploit the results
HEAP Consortium: Synelixis

• Relevant Expertise
  – FPGA-based highly parallel systems, real-time Reconfigurable systems for various applications.

• Main Project Role
  – Technical Coordination of the project
  – Contribute to the integration of the parallel platform
• Relevant Expertise
  – Design, implementation and marketing of high-end embedded systems, supporting, among others, real-time video streaming and/or image processing.

• Main Project Role
  – Provide requirements for the parallel framework in different domains
  – Provide embedded highly CPU intensive applications
  – Evaluate and exploit the results
• Relevant Expertise
  – Design and implementation of highly flexible, easy-targetable compiler system, creating high-quality, high-performance compilers for a broad spectrum of DSP, NPU, RISC, VLIW, 8/16-bit microcontroller and multi-core architectures.

• Main Project Role
  – Provide requirements for parallelizing tools
  – Contribute to the implementation of the parallelizing tool
  – Tailor toolset to the COSY compiler
  – Evaluate and exploit the results
• Relevant Expertise
  – Design and implement innovative products to program heterogeneous multi-core systems from high-level programming languages like Matlab or C.

• Main Project Role
  – Provide requirements for parallelizing tools
  – Contribute to the implementation of the parallelizing tool
  – Evaluate and exploit the results
HEAP Consortium: Politecnico Di Torino

• Relevant Expertise
  – Design and implementation of software tools for high-level synthesis and for parallelizing sequential code, and verification tools for parallel and hardware/software embedded systems.

• Main Project Role
  – Main responsible for the design and implementation of the parallelizing software including the verification tools
  – Evaluate and exploit results with regards to verification coverage
• Relevant Expertise
  – Design and implementation of numerous cache coherency protocols including the most efficient such protocol worldwide.

• Main Project Role
  – Specify, design and implement the numerous HEAP Cache Coherency protocols
HEAP Consortium:
Universita Degli Studi Di Genova

• Relevant Expertise
  – Design and implementing highly parallel applications for various application domains executed on GPUs, multicore, and custom parallel architectures.

• Main Project Role
  – Provide high-end applications, parallelizing them manually and using third-party tools
  – Main responsible for the evaluation set up
HEAP Consortium: Singular Logic

• Relevant Expertise
  – Highly parallel systems, including both the software and hardware aspects of them.

• Main Project Role
  – Main responsible for implementing the parallel system
  – Contribute heavily in the dissemination and exploitation plan
HEAP Consortium: Uppsala Universitet

• Relevant Expertise
  – Design and implementation of numerous cache coherency protocols.

• Main Project Role
  – (Uppsala Universitet will undertake part of ATHENA’s work in the project)
HEAP Funding

- Duration of the project 33 months
- EC Financial Contribution: €2,216,069
- Total Cost: €3,319,534

Funding Distribution per Partner

- Athena 10%
- UniGe 12%
- STM 14%
- Synelaxis 13%
- PoliTo 19%
- Thales 14%
- Compaan 9%
- ACE 9%

* Uppsala Universitet not present in the charts. Funds from ATHENA will be transferred to Uppsala
HEAP Technical: Architecture of the tool set

- Profiler
- Annotated C/C++
- Execution
- Data traces
- Constraint solver
- Front-end
- Int. Rep.
- Visualization GUI
- Compressor
- Parallel C/C++
- Output
- Polyhedral analysis

Sequential C/C++
Main aim is to hide from the programmer the intricacies of the cache coherence protocols.

- Use information from the parallelizing process and tailor the CC protocol to the application.

Optimization will focus on both performance and energy consumption.

- Cache coherence optimizations will consider energy reductions (e.g., reduction in network traffic, cache accesses, etc) in addition to (or in parallel to) more traditional latency optimizations.
- Emphasis can be shifted towards what matters most.
Situations where coherence related optimizations can apply:

- Optimizing the application for a given cache architecture
  - Issues such as false sharing, communication efficiency, invalidation traffic, coherence traffic bursts, highly-contended data, can be dealt with using appropriate data partitioning, placement, replication, etc. The code and data transformations required in this case are specific to the targeted cache architecture. A variation in this case, is to **identify the best cache architecture for an application, in a given design space.** This involves the evaluation of the behaviour of the application for each design point, given the available profiling, formal analysis, and parallelization information.

- Customizing cache coherence to the application
  - Starting from the minimalistic approach of Writer Coherence, we will provide a coherence substrate on top of which we can develop efficient and customizable cache coherence modules tailored to application needs. In many such protocols the applications themselves will be able to directly drive coherence operations. **Coherence protocols existing today are application-agnostic,** that is, they do not exploit knowledge of application access patterns. Using the dependence and parallelization information available in the toolset, coherence protocols can be designed specifically for the applications.
<table>
<thead>
<tr>
<th>Workpackage No</th>
<th>Workpackage title</th>
<th>Type of Activity</th>
<th>Lead partic. no</th>
<th>Lead partic. short name</th>
<th>Person - months</th>
<th>Start month</th>
<th>End month</th>
</tr>
</thead>
<tbody>
<tr>
<td>WP1</td>
<td>Project Coordination &amp; Management</td>
<td>MGT</td>
<td>1</td>
<td>STM</td>
<td>22.5</td>
<td>M1</td>
<td>M33</td>
</tr>
<tr>
<td>WP2</td>
<td>Requirements Capture &amp; Specifications</td>
<td>RTD</td>
<td>3</td>
<td>STM</td>
<td>34.5</td>
<td>M1</td>
<td>M7</td>
</tr>
<tr>
<td>WP3</td>
<td>Implementation of Parallelizing Toolset</td>
<td>RTD</td>
<td>6</td>
<td>PoliTo</td>
<td>99</td>
<td>M6</td>
<td>M26</td>
</tr>
<tr>
<td>WP4</td>
<td>Implementation of Parallel Hardware System</td>
<td>RTD</td>
<td>7</td>
<td>ATHENA</td>
<td>62</td>
<td>M6</td>
<td>M24</td>
</tr>
<tr>
<td>WP5</td>
<td>Integration Demonstration and Evaluation</td>
<td>RTD</td>
<td>8</td>
<td>Thales</td>
<td>97</td>
<td>M16</td>
<td>M33</td>
</tr>
<tr>
<td>WP6</td>
<td>Exploitation and Dissemination</td>
<td>DEM</td>
<td>1</td>
<td>Thales</td>
<td>23</td>
<td>M3</td>
<td>M33</td>
</tr>
</tbody>
</table>
### HEAP Work Structure: Pert Chart

**1.1 Project Co-ordination**

- WP 1. Project Coordination & Management
  - 2.1: Requirements capture and analysis
  - 2.2: Specify architecture of toolset
  - 2.3: Cache Coherency modules
  - 2.4: Parallel System specifications
  - 2.5: Specify Microarchitecture of Hardware Modules and High-level Architecture

**3.1: Profiling-based optimistic data dependency analysis**

**3.2: Array-based conservative data dependency analysis**

**3.3: Dependency visualization and code decomposition tool**

**3.4: Code generation back end**

**3.5: Metric-driven verification**

**3.6: HEAP framework functional verification**

**WP3. Implementation of Parallelizing Toolset**

- 3.1: Profiling-based optimistic data dependency analysis
- 3.2: Array-based conservative data dependency analysis
- 3.3: Dependency visualization and code decomposition tool
- 3.4: Code generation back end
- 3.5: Metric-driven verification
- 3.6: HEAP framework functional verification

**WP4. Implementation of Parallel Hardware System**

- 4.1: Design Cache coherency
- 4.2: Implementation of Cache coherency modules
- 4.3: Design, integrate and implement the parallel system
- 4.4: Logic synthesis, timing verification
- 4.5: Validation of I/O and memory throughput

**WP5. Demonstration & Evaluation**

- 5.1: Trials Specification & Set-up
- 5.2: Implementation of Demonstration Applications
- 5.3: HEAP integration
- 5.4: Evaluation of the HEAP framework

**WP6. Exploitation & Dissemination**

- 6.1 Market Assessment & Analysis
- 6.2 Exploitation Plan & contribution to the standards
- 6.3 Dissemination, Workshops

---

**Exploitation/Dissemination for Specific Tasks or complete WPs**

- WP Input
- Specific Task Input
• Technical Achievements:
  – The specifications of the toolset and architecture have been identified, analysed and delivered on time.
  – The specifications of the cache coherency modules have been identified, analysed and delivered on time.
  – The specifications and high-level architecture of the parallel system have been identified, analysed and delivered on time.
  – The first version of the specifications of the data dependencies model has been identified, analysed and delivered on time.
  – The cache coherency modules have been designed and delivered on time.

• Knowledge Dissemination:
  – Web site online (www.fp7-heap.eu) – makes available a number of public documents
  – HEAP partners have published papers in refereed conferences/workshops and made a number of poster presentations
• Project Liaison:
  – HEAP has not started any liaison activities with other projects yet, except of the meeting and presentation to the HiPEAC NoE corresponding cluster.
Achievements for WorkPackage 1 (in progress):

- Provided effective project management at all stages throughout the duration of the project.
- Initiated all necessary administrative tasks and provided regular progress reports to the Commission.
- Supervised the technical progress of the project.
- Ensured the signing by all partners of the consortium agreement, handling among others the IPR issues.
- Established and maintained effective communication between project participants.
- Ensures a proper project administration and coordination.
- All those achievements are described in the Project handbook.
Achievements for WorkPackage 2 (completed):

• The requirements for the parallelizing toolset, the cache coherency protocols and the hardware system were captured and defined

• Specifications for all major parts of the project have been completed
  – work now proceeds in parallel with partners working simultaneously on the parallelizing toolset, cache coherency protocols and hardware system development.

• All related deliverables have been completed in time and reached a level of quality that satisfies all partners.
Achievements for WorkPackage 3 (in progress):

- All the specifications that will ensure smooth interfacing and interoperability between profiling and display have been written.
- A prototype of the profiling code and of the associated runtime profiling library have been completed.
Achievements for WorkPackage 4 (in progress):

• Detailed description of two sets of cache coherency modules (the first set based on directory coherence and to be used in the HEAP high-performance multi-core system, while the second set based on snooping coherence and to be used in the HEAP low-power embedded multi-core system).

• For each set, two extra variations of the baseline protocol are also described.

• For every protocol, we provide its state transition diagrams which fully describe the functionality of the cache, memory and directory controllers.

• The hardware requirements of each protocol are also specified.

• For each coherency module, a detailed description of the required input needed by the HEAP toolset is also specified.
Beyond HEAP: Exploitation Plans

- HEAP is considered of great significance for all the participating companies and academic centers
  - Exploitation plans have been made and are updated as the project progresses
  - Each partner has an exploitation plan and academic/commercial dissemination strategy
Beyond HEAP: Exploitation Plans (cntd)

• Commercial Partners
  – Plan to evolve their current product roadmap by using the expertise and IP developed in the project to progress their products towards a more usable multiprocessing framework
  – Seek to map complex applications (graphics, multimedia) into HEAP framework to reduce time-to-market and gain competitive advantage
  – Commercial dissemination plan includes training of internal research and product development teams as well as demonstration of the platform in customers and partners through workshops, trade shows and exhibitions
Beyond HEAP: Exploitation Plans (cntd)

• Academic Partners
  – Will apply knowledge gained from HEAP in other research projects they participate as well as in the consulting services they provide
  – Plan to use HEAP’s tools and experience in graduate courses with collaborating universities
  – Academic dissemination plans include publications in conferences and journals, where a number of papers has already been accepted
  – Explore opportunities for spin-offs/start-ups and use HEAP’s knowledge/experience in industrial consulting
Thank You!

Andreas Brokalakis
Synelixis Solutions Ltd
brokalakis@synelixis.com