1 Elementary Notions of Heterogeneous Computing

1.1 Description

This module introduces fundamental concepts in heterogeneous computing. Notions of concurrency, parallelism, and energy efficiency are discussed to explain the motivation behind the move towards heterogeneous processing. Different forms of heterogeneity are introduced including soft heterogeneity (i.e., difference in core compute capabilities within a multicore system), CPU-GPU heterogeneous execution and System-on-Chip (SoC) design. The module also covers heterogeneity in workload and data with examples from cloud computing and mobile applications. The module concludes with a discussion of programmability and performance challenges.

1.2 Context

This module is primarily intended for CS1/CS2 students. Although the module introduces parallel computing concepts before moving on to processor heterogeneity, it is ideally suited for a course with some coverage of parallel computing material. For example, a CS1 course that incorporates a PDC module from [1], [2], or [3]. In the absence of PDC coverage, the length of this module will need to be increased or it will need to be combined with a PDC module.

1.3 Topics

The HC topics covered in this module are listed below. Bloom's classification is shown in brackets

- Concurrency and Parallelism [K]
- Multicore Processors [K]
- GPGPU [K]
- System-on-Chip (e.g., mobile processors)
- Energy Efficiency [K]
- Tasks and Workloads [K]
- Task Mapping and Scheduling [K]
- Amdahl's Law [C]

1.4 Learning Outcomes

Having completed this module, students should be able to

- describe the differences between a homogeneous and heterogeneous computing system

- describe and distinguish between different forms of heterogeneity
- understand the motivation behind the design of heterogeneous computing systems
- recognize the importance of energy efficiency on current computing systems
- understand that tasks in a workload have different demands for compute and memory resources
- understand the notion of task mapping as performed by an operating system
- analyze the performance and energy effects of task mapping on a heterogeneous system

1.5 Pedagogical Notes

To be updated.

1.6 Instructor Resources

The teaching material included with this module include the following

- Lecture slides: includes notes on guiding class discussion
- In-class Demo: includes instructions for setting up a heterogeneous environment within a homogeneous multicore system and step-by-step guidelines for running the demo in class
- Lab : The module includes a lab that provides students hands-on experience in running application on a heterogeneous system. The lab will also reinforce the performance and energy implications covered in the lecture. The lab includes detailed instructions for the instructor in setting up a heterogeneous system on which students will conduct performance experiments. The lab requires the student to have some basic familiarity with a Linux environment.
- Reference Material: further reading for instructors unfamiliar with topics covered in this module
- Pedagogical Notes: suggestions drawn from author's own experience in teaching this module

All material is available for download from https://github.com/TeachingUndergradsCHC/modules.git under folder moduleA.

References

[1] CSinParallel Project. `http://csinparallel.org/`.

[2] Center for parallel and distributed computing curriculum development and educational resources (CDER). `http://www.cs.gsu.edu/~tcpp`.

[3] Parallel Computing in the Undergraduate Curriculum : the Early-and-Often Approach. http://tues.cs.txstate.edu/.