Bahurupi: Polymorphic Heterogeneous Multi-Core Systems

PI: Dr. Tulika Mitra [School of Computing, National University of Singapore]
External Collaborator: Prof. Sri Parameswaran [University of New South Wales, Australia]

Recently, there has been an exponential growth in the number of applications whose computational demands far exceed the capabilities of even the most powerful general-purpose computing platforms, such as commodity clusters or general-purpose scientific supercomputers. These futuristic applications, which include bio-informatics, physical simulation, computer vision, financial analysis, data mining, etc., are important emerging workloads for computing systems. So it is essential to architect computing platforms that can enable these applications to achieve the desirable performance level. As an example, accurate molecular dynamics (MD) simulations involving protein, cell membranes, and DNA allow scientists and drug designers to visualize many critical biochemical phenomenons. IBM Blue Gene/L — the most powerful supercomputer as of today employing 4,096 processing cores — will take an astronomical 210 years to simulate (at simulation rate of 13ns/day) just one millisecond worth of a biochemical process. Unfortunately, scientists start to observe most of the biologically interesting phenomenon only at millisecond timescale. Clearly, we need at least three orders of magnitude speedup beyond what is currently offered by state-of-the-art computing machinery.

At the systems level, the emergence of multi-cores and eventually many-cores (multiple processing cores or CPU on a single chip) has brought the computer community to a crossroad. Desktop and laptops have already made the irreversible transition toward multi-cores due to thermal/power constraints, reliability issues, design complexity, and so on. As transistor density continues to enjoy the exponential growth according to Moore's Law, the number of on-chip cores (but not the performance) is predicted to double every two years. However, we cannot ride this growth curve for on-chip cores to obtain the kind of performance speedup we are looking for in the near future.

The current commercial trend is to build multiprocessors that are just collections of identical (possibly simple) cores. These homogeneous multi-cores are simple to design, offer easy silicon implementation, and regular software environments. Unfortunately, general-purpose emerging workloads from diverse application domains have very different resource requirements that are hard to satisfy with a set of identical cores. In contrast, there exist many evidences that heterogeneous multi-core solutions customized for a particular application domain can offer significant advantage in terms performance, power, area, and delay. Even though customization is a promising option, it is not practical for general-purpose workloads due to high non-recurring engineering cost (NRE). First, designing customized heterogeneous multi-cores from scratch for every new and upcoming important application is a challenging proposition. More importantly, custom fabrication of each new design to realize the performance benefit is a viable option only if it is economically sustainable by a large market (which might be feasible in the embedded domain but is hard to justify for general-purpose applications that we are targeting). Can we possibly leverage the performance boost offered by customization without incurring high NRE cost?

In this project, we propose a polymorphic heterogeneous multi-core architecture, named Bahurupi (a Bengali word meaning a person of many forms and guises, a polymorph), that can be tailored according to the workload by software. Bahurupi will be designed and fabricated as a heterogeneous multi-core system containing multiple identical (simple) cores and memory modules, a number of specialized processing engines (e.g., FPUs, vector processors, DSPs etc.), as well as some amount of re-configurable logic on chip. The main novelty of Bahurupi lies in its highly flexible architecture. Post-fabrication, software can configure or compose together primitive on-chip hardware components to create a customized multi-core system that best matches the needs of a specific application. Put simply, Bahurupi bypasses the costly design and fabrication challenges of customized multi-cores through software directed re-configurability.

Bahurupi will allow post-fabrication, software directed configuration and customization along multiple dimensions. First, we propose to design composable cores that can be fused together at runtime to create larger cores so as to match the number of cores with the number of threads and possibly perform heavy-lifting of the sequential portion of the computation. Similarly, the primitive memory modules can be composed together, under software directives, in a variety of ways to deliver a memory subsystem that suits the resource requirements and sharing characteristics of the workload. The cores are expected to share specialized processing units, such as FPUs, vector processors etc., to save real estate. Finally, we would like to extend, again at runtime, the instruction-set architecture (ISA) of the processing core with application-specific instructions (corresponding to the kernel running on it) implemented in the re-configurable fabric on-chip.

The success of Bahurupi will be largely determined by the availability of quality software tool-chain that can identify the appropriate configuration for a particular workload. We expect the software developer to expose data parallelism, task parallelism, and pipelined parallelism in the application, i.e., we do not plan to address the automated parallelization issue. Given an application with exposed parallelism, we propose to develop an efficient compile-time technique that can quickly evaluate the large design space and come up with a Pareto Front containing a set of promising configurations along multiple dimensions — performance, power, and thermal behavior. Once the application developer selects a configuration from this Pareto Front, we will offer support for an efficient mapping of the application to the configured architecture.

In summary, the goal of this project is to (a) design a polymorphic heterogeneous multi-core system that can be configured, under software control, to match the requirement of the important emerging applications, and (b) design an appropriate tool chain that help the software developer to identify and map an application to the appropriate configuration.