Thrifty: An Exascale Architecture for Energy-Proportional Computing

Josep Torrellas (University of Illinois), Daniel Quinlan (Lawrence Livermore National Laboratory),
Allan Snavely (University of California, San Diego), Wilfred Pinfold (Intel)

1 Context

Current petascale architectures are not scalable. They consume several megawatts of power and can easily waste 20% of their capacity to faults and recovery. As we look toward building exascale machines, we need to make major strides in power/energy-efficiency, resiliency, and performance. To address these fundamental challenges, this work proposes a novel exascale architecture called *Thrifty*. Thrifty, together with the research thrusts proposed, aims to provide a platform for highly-efficient, energy-proportional computing — a platform where energy is spent proportionally to the work done, and in the resources that truly are the bottleneck.

The Thrifty concept and its software stack simultaneously innovate in power/energy efficiency, resiliency, and performance. Each topic is tackled at multiple layers of the computing stack: circuits, architecture, compilation, and runtime/application.

2 Objectives

Power/Energy Efficiency: Attain orders-of-magnitude efficiency gains over current systems by developing:

- Novel circuits and architecture technologies for low supply voltage, and fine-grain management of static and dynamic power.
- A power/energy-aware source-to-source compiler that uses static and dynamic code analysis to drive the power-management hardware and auto-tunes code for power/energy efficiency.
- Application models for efficient energy-proportional computing and a means to insert power-management pragmas in the application.

Resiliency: Reduce the waste due to faults and recovery to less than 5% of execution by developing:

- New circuits and architecture technologies for high resiliency at low supply voltage and efficient error detection and tolerance.
- A novel architectural scheme for energy-efficient, incremental, in-memory, scalable checkpointing.
- Compilation and applications support to drive the checkpointing scheme.

Performance: Attain orders-of-magnitude performance increases over current machines by:

- Architecting many performance-enhancing features in Thrifty, including primitives for fine-grain synchronization and communication.
- Developing a compiler that efficiently drives the performance features of Thrifty and auto-tunes programs for performance.
- Identifying application *Idioms* and mapping them efficiently on Thrifty.

3 Thrifty Architecture

Thrifty is an exascale architecture that is designed for exascale DoE applications such as S3D, Madness, POP, WRF, and GTC. The building block is a many-core chip with 1024 cores. Processor and memory chips are organized in MCMs, boards and cabinets. The system in a cabinet shares a single address space. The research in this project focuses on the design and evaluation with simulations of only a 1K-core machine. Figure 1 shows a picture of Thrifty.

Processors and memory hierarchies operate at a low supply voltage close to the transistors' threshold voltage. There is hardware and software support for fine-grain management of the on-chip dynamic and static power.

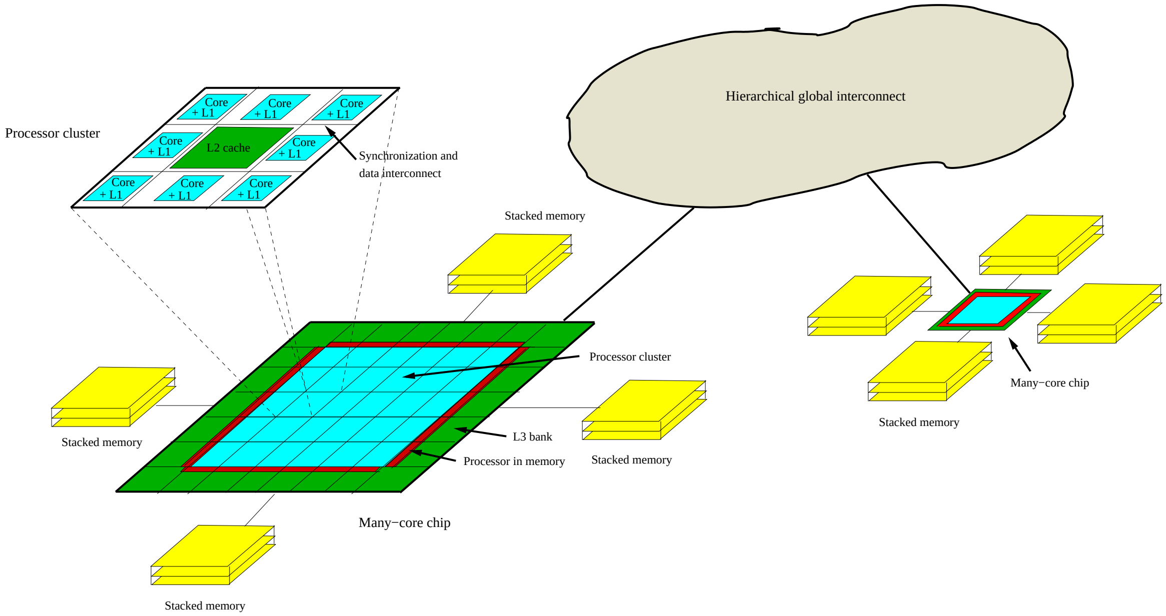

Figure 1: The Thrifty exascale architecture.

The cores are narrow-issue, in-order, and have units that are reconfigurable and can be shut down. Cores are organized in groups called Clusters. The cores in a cluster share a level of the memory hierarchy and work together in a fine-grained manner. They have special hardware that enables fine-grain synchronization and communication.

All the clusters in a chip share another level of memory hierarchy. Such a memory is banked and distributed across the chip. Each bank or group of banks has a small hardware unit (Processor in Memory or PIM) that performs specialized memory-intensive operations.

The chip uses a reconfigurable network with wide links to deliver high bandwidth, and simple routers to ensure low-power operation.

The chip contains extensive hardware for resiliency. This includes error-correcting codes, parity bits in data paths and network links, testing circuitry exercised periodically, and a variety of sensors. The system also includes a novel low-overhead, in-memory incremental check-pointing scheme that leverages non-volatile memory.

Thrifty includes the concept of an engine to perform block transfers across different levels of the memory hierarchy. The chip also provides mechanisms based on hardware address signatures to detect data transfer patterns and minimize, hide, and reduce data movement. Finally, it includes counters and probes to measure power and other parameters for code auto-tuning.

A chip has connections to off-chip stacked memory. Four processor chips are combined into an MCM. Then, four MCMs and substantial memory are placed on a board. The global interconnect is hierarchical, organized in a fat tree and implemented in a combination of electrical and optical interconnect. Many of the on-chip architectural features are extended hierarchically to work over bigger parts of the machine.

4 Scope

The work is proposed by an interdisciplinary team of researchers from academia, national laboratories and industry — each expert on a different layer of the computing stack. The work involves the development of an architecture simulator of a 1K-core Thrifty system and its evaluation, the implementation and evaluation of a power-aware optimizing compiler based on ROSE, the development of libraries for application characterization and tuning, and the fabrication of a test chip to test some of the technologies.