Exploring Manycore Multinode Systems for Irregular Applications with FPGA Prototyping

Marco Ceriani, Simone Secchi, Antonino Tumeo, Oreste Villa and Gianluca Palermo

Pacific Northwest NATIONAL LABORATORY
Proudly Operated by Battelle Since 1965

Motivation

New knowledge discovery applications, such as social network analysis, employ big pointer-based data structures, including graphs or trees.

They have large amounts of latent parallelism, but the memory patterns are highly irregular.

Current High Performance Computing (HPC) systems employ processors with advanced cache hierarchies. They are optimized for regular computations and scientific applications with high temporal and spatial locality. Conversely, irregular applications have poor or no locality. In addition, their datasets are very difficult to partition in a balanced way.

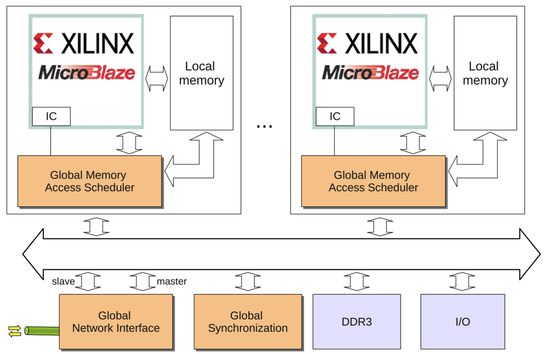
Features for irregular applications

- Global address space: eases programming of the applications
- Scrambling of the address space: reduces network hotspots
- Fine-grained multithreading (software or hardware): tolerates data access latencies
- Fine-grained synchronization

Our approach

Speed up the execution and development of irregular applications on commodity multi-core architectures with limited additional custom hardware.

The target is designing more cost-effective HPC clusters capable of efficiently supporting both regular and irregular applications.



Global address space

The data port of the processor is monitored to identify and transform all memory operations.

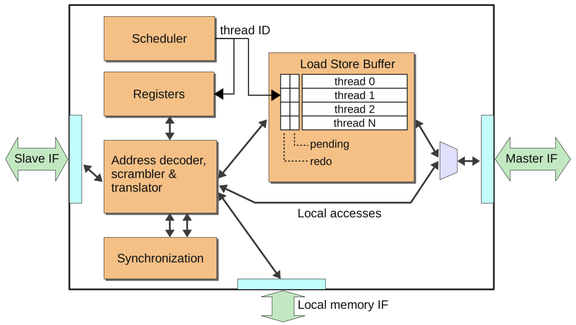
- Configurable scrambling of addresses in global space
- Local accesses forwarded without penalty
- Remote accesses sent to the network interface
- Memory transactions are transparently encapsulated in network packets.

Thread scheduling

- Remote accesses immediately trigger a context switch by raising an interrupt.
- The hardware scheduler applies round-robin priority over threads with no pending accesses.

Prototype features

- Implemented on FPGA
- Based on off-the-shelf soft-core processors
- Global address space distributed on multiple nodes
- Scrambled global addresses
- Lightweight software multi-threading supports integration with the global memory access scheduler
- Fine-grained synchronization primitives to protect many small critical sections



Fine-grained synchronization

Lock/unlock operations on memory addresses provide synchronization at word granularity.

The support is implemented with a memory-mapped lock table included on every node.

Components occupation

Occupation on a Virtex-6 LX240T FPGA

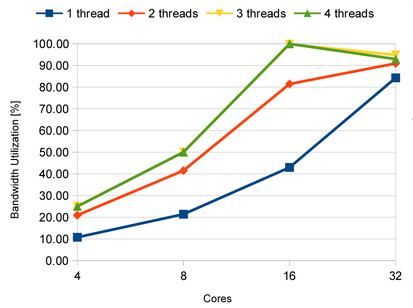
	# of slice registers	# of slice LUTs	$f_{max}[MHz]$
GMAS	1063 (0.3%)	1566 (1.0%)	230.3
GNI	450 (0.1%)	1915 (1.3%)	253.3
GSYNC	4711 (1.6%)	8054 (5.3%)	241.1
Switch	92 (0%)	589 (0.4%)	237.9

Prototype evaluation

Architecture:

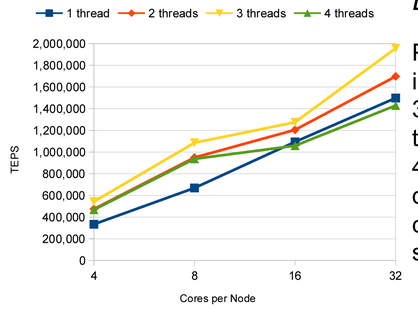
- 4 Xilinx ML605 boards, each mounting a Virtex-6 LX240T FPGA
- Fully interconnected mesh with GTX transceiver (3 x board, 500 Mbit/s each)
- Total bandwidth per board: 1.5 Gbit/s
- From 1 to 32 cores per FPGA, 1 to 4 threads per core
- Lock tables: 8096 entries
- Thread switching latency: 232 cycles
- 512 MB DDR3 RAM per FPGA (32 MB local)
- Round trip time: 403 cycles.

Benchmarks: pointer chasing, breadth first search (BFS)



Pointer Chasing

Increasing cores and threads maximizes bandwidth utilization.



BFS

Performance increases: up to 32 cores, up to 3 threads.
4 threads: slow down, due to contention, synchronization.

Marco Ceriani; Gianluca Palermo
Politecnico di Milano, DEIB
mceriani@elet.polimi.it; gpalermo@elet.polimi.it

Simone Secchi
Università di Cagliari, DIEE
simone.secchi@diee.unica.it

Antonino Tumeo; Oreste Villa
Pacific Northwest National Laboratory, HPC
antonino.tumeo@pnnl.gov; oreste.villa@pnnl.gov

U.S. DEPARTMENT OF ENERGY
www.pnnl.gov
PNNL-SA-90226