A Framework for the Automatic Vectorization of Parallel Sort on x86-based Processors

Abstract—The continued growth in the width of vector registers and the evolving library of intrinsics on the modern x86 processors make manual optimizations for data-level parallelism tedious and error-prone. In this paper, we focus on parallel sorting, a building block for many higher-level applications, and propose a framework for the Automatic SIMDization of Parallel Sorting (ASPaS) on x86-based multi- and many-core processors. That is, ASPaS takes any sorting network and a given instruction set architecture (ISA) as inputs and automatically generates vector code for that sorting network. After formalizing the sort function as a sequence of comparators and the transpose and merge functions as sequences of vector-matrix multiplications, ASPaS can map these functions to operations from a selected “pattern pool” that is based on the characteristics of parallel sorting, and then generate the vector code with the real ISA intrinsics. The performance evaluation on the Intel Ivy Bridge and Haswell CPUs, and Knights Corner MIC illustrates that automatically generated sorting codes from ASPaS can outperform the widely used sorting tools, achieving up to 5.2x speedup over the single-threaded implementations from STL and Boost and up to 6.7x speedup over the multi-threaded parallel sort from Intel TBB.

CONCLUSION

In this paper, we propose the ASPaS framework to automatically generate vectorized sorting code for x86-based multicore and manycore processors. ASPaS can formalize the sorting and merging networks to the sequences of comparing and reordering operators of DSL. Based on the characteristics of such operators, ASPaS first creates an ISAfriendly pool to contain the requisite data comparing and reordering primitives, then builds those sequences with primitives, and finally maps them to the real ISA intrinsics. Besides, the ASPaS codes can exhibit a efficient memory access pattern and thread-level parallelism. The ASPaS generated
codes can outperform the compiler-optimized ones and meanwhile yield highest performance over multiple library sorting tools on Ivy Bridge, Haswell, and Knights Corner architectures.

SYSTEM REQUIREMENTS:

HARDWARE REQUIREMENTS:

- System: Pentium IV 2.4 GHz.
- Hard Disk: 40 GB.
- Floppy Drive: 1.44 Mb.
- Monitor: 15 VGA Colour.
- Mouse: Logitech.
- Ram: 512 Mb

SOFTWARE REQUIREMENTS:

- Operating system: Windows 7/UBUNTU.
- Coding Language: Java 1.7, Hadoop 0.8.1
- IDE: Eclipse
- Database: MYSQL

REFERENCES
