

Researchers use hardware to accelerate core-to-core on-chip communication

September 6 2016, by Matt Shipman

Researchers from North Carolina State University and the Intel Corporation have developed a new way to significantly accelerate core-to-core communication. Their advance relies on hardware to coordinate efforts between cores for multiprocessor operations.

Many computer functions require multiple processors, or cores, to work together in a coordinated way. Currently, this coordination is achieved by sending and receiving software commands between cores. But this requires cores to read and execute the software, which takes time.

Now researchers have developed a chip design that replaces the software instructions with built-in hardware that coordinates [communication](#) between cores, accelerating the process.

"This approach, called the [core](#)-to-core communication acceleration framework (CAF), improves communication performance by two to 12 times," says Yan Solihin, a professor of electrical and computer engineering at NC State and co-author of a paper on the work. "In other words, the execution times - from start to finish - are twice as fast or faster."

The key to the CAF design is a queue management device (QMD), which is a small device attached to the processor network on a chip. The QMD is capable of simple computational functions and effectively keeps track of communication requests between cores without having to rely on [software](#) routines.

The researchers have also found that, because it can perform basic computation, the QMD can be used to aggregate data from [multiple cores](#) - expediting some basic computational functions by as much as 15 percent.

"We are now looking at developing other on-chip devices that could accelerate more multi-core computations," Solihin says.

The paper, "CAF: Core to Core Communication Acceleration Framework," will be presented at the 25th Annual Conference on Parallel Architectures and Compilation Techniques, being held Sept. 11 to 15 in Haifa, Israel.

More information: "CAF: Core to Core Communication Acceleration Framework" the 25th Annual Conference on Parallel Architectures and Compilation Techniques, 2016.

Abstract

As the number of cores in a multicore system increases, core-to-core (C2C) communication is increasingly limiting the performance scaling of workloads that share data frequently. The traditional way cores communicate is by using shared memory space between them. However, shared memory communication fundamentally involves coherence invalidations and cache misses, which cause large performance overheads and incur a high amount of network traffic. Many important workloads incur significant C2C communication and are affected significantly by the costs, including pipelined packet processing which is widely used in software-based networking solutions. In these workloads, threads run on different cores and pass packets from one core to another for different stages of processing using software queues. In this paper, we analyze the behavior and overheads of software queue management. Based on this analysis, we propose a novel C2C Communication Acceleration Framework (CAF) to optimize C2C communication. CAF

offloads substantial communication burdens from cores and memory to a designated, efficient hardware device we refer to as Queue Management Device (QMD) attached to the Network on Chip. CAF combines hardware and software optimizations to effectively reduce the queue-induced communication overheads and improve the overall system performance by up to 2-12x over traditional software queue implementations.

Provided by North Carolina State University

Citation: Researchers use hardware to accelerate core-to-core on-chip communication (2016, September 6) retrieved 20 March 2024 from <https://phys.org/news/2016-09-hardware-core-to-core-on-chip.html>

<p>This document is subject to copyright. Apart from any fair dealing for the purpose of private study or research, no part may be reproduced without the written permission. The content is provided for information purposes only.</p>
--