ATAC: Improving performance and programmability

ATAC: Improving Performance and Programmability with On-Chip Optical Networks

James Psota, Jason Miller, George Kurian, Henry Hoffman, Nathan Beckmann, Jonathan Eastep, Anant Agarwal
Massachusetts Institute of Technology, Cambridge, MA

***Abstract*—Given the current trends in multicore scaling, chips with 1000 cores may exist within the next 5 to 10 years. However, their promise of increased performance will only be reached if their inherent scaling and programming challenges are overcome. Meanwhile, recent advances in nanophotonic device manufacturing are making CMOS-integrated optics a reality—interconnect technology which can provide more bandwidth at lower power than conventional electronics. Perhaps more importantly, optical interconnect also has the potential to enable new, easy-to-use programming models enabled by its inexpensive broadcast mechanism.**

This paper introduces ATAC, a new manycore architecture that capitalizes on the recent advances in optics to address a number of challenges that future manycore designs will face. The new constraints and opportunities of on-chip optical interconnect are presented and explored in the design of ATAC. Furthermore, this paper discusses ATAC's programming models, and introduces Consumer Tagging, a novel programming model that leverages ATAC's strengths to provide high performance and scalability.

I. Introduction

As silicon resources become increasingly abundant, massively multicore chips are on the horizon. This paper illustrates the scalability challenges faced by current multicore architectures and how ATAC addresses them. ATAC integrates an on-chip optical broadcast communication network within a mesh based tiled multicore architecture to significantly improve the performance, energy scalability, and ease of programmability of multicore processors [1].

Current multicores typically employ either bus-based interconnect (for small number of cores) or a mesh-based interconnect (for large number of cores). While more scalable than the bus approach, the mesh approach is highly prone to contention as processors scale to thousands of cores. Furthermore, multicore architectures are threatened by the programming challenge, as programmers must orchestrate computation and communication. ATAC addresses these issues by integrating on-chip optical communication technologies to augment electrical communication channels. ATAC virtually eliminates communication contention using Wavelength Division Multiplexing (WDM), allowing a single waveguide to simultaneously carry multiple independent signals on different wavelengths. The optical waveguide consumes lower energy and has lower transmission latency over long distances relative to an electrical mesh network. It also enables uniform communication between any pair of cores, regardless of distance, which improves programmability.

This paper gives an overview of the ATAC architecture, shows how the broadcast network can be leveraged to improve multicore programming, and evaluates ATAC versus an electrical-only multicore. ATAC also enables ACKwise, an efficient new cache coherence protocol, which is outside the scope of this paper and is described in [2].

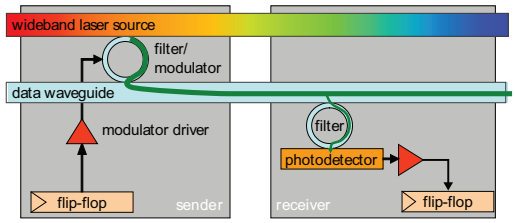

Fig. 2. Optical transmission of a single bit

II. Optical Technology Background

This section gives a brief overview of the optical components used in ATAC. The key elements in a nanophotonic network such as the one employed by the ATAC chip include: the offchip "optical power supply" light source; waveguides to carry optical signals; filters and modulators to place signals into the waveguides; and detectors to receive signals from the waveguides. Figure 2 puts all of these elements together, showing how one bit is transmitted from a flip-flop of one core to a flip-flop of another core. The core on the left shows the components relevant to sending and the core on the right shows the components relevant to receiving. The process for sending a bit on the ATAC's optical network is as follows. The flip-flop signals the modulator driver to send either a 0 or a 1. The modulator driver, which consists of a series of inverter stages, drives the modulator's capacitive load. The modulator couples light at its pre-tuned wavelength λ_i from the optical power source and encodes either a 0 or 1 onto the data waveguide. The optically-encoded data signal traverses the waveguide at approximately one-third the speed of light and is detected by a filter that is also tuned to wavelength λ_i. Photons are detected by the photodetector and received by a flip-flop on the receiver side. Further detail about ATAC's optics is available in [2].

III. Architecture Overview

The ATAC architecture is a tiled multicore architecture combining the best of current scalable electrical interconnects with cutting-edge on-chip optical communication networks. The tiled layout uses a 2-D array of simple processing cores, each containing a single- or dual-issue, in-order RISC pipeline, and L1 data and instruction caches. The ATAC architecture is targeted at an 11nm process in 2019, and will have at least 1000 cores (here we assume 1024).

The cores in an ATAC processor are connected through two networks: the electrical EMesh and the optical/electrical ANet. The EMesh is a conventional 2-D point-to-point electrical mesh network and is ideal for predictable, short-range communication. The ANet employs state-of-the-art optical technology to enable low-latency,

978-1-4244-5309-2/10/$26.00 ©2010 IEEE

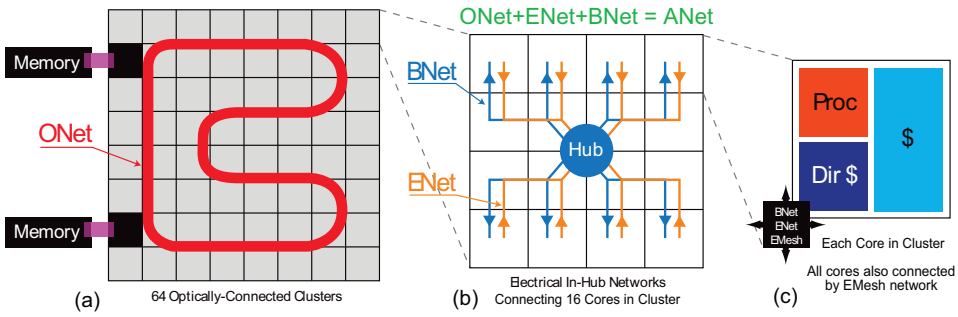

Fig. 1. ATAC architecture overview

energy-efficient, contention-free global communication. The core of the ANet is the all-optical ONet shown in Figure 1. The ANet also contains two small electrical structures called the ENet and BNet that are used to interface with the ONet. The ANet is especially useful for long-distance communication or global operations such as broadcasts. The remainder of this paper focuses on the ANet.

The ONet provides a low-latency, contention-free connection between a set of optical endpoints called Hubs. Hubs are interconnected via waveguides that visit every Hub and loop around on themselves to form continuous rings (see Figure 1). Each Hub can place data onto the waveguides using an optical modulator and receive data from the other Hubs using optical filters and photodetectors. Because the data waveguides form a loop, a signal sent from any Hub will quickly reach all of the other Hubs. Thus every transmission on the ONet has the potential to be a fast, efficient broadcast.

To avoid the interference of these broadcasts with each other, the ONet uses wavelength division multiplexing (WDM). Each Hub has modulators tuned to a unique wavelength to use when sending and contains filters that allow it to receive signals on all the wavelengths. This eliminates contention and the need for arbitration in the optical network. In addition, the improved propagation speed of optical signals eliminates the heterogeneous, distance-dependent cost of communication between cores; any pair of Hubs on the chip can communicate with low, fixed latency instead of the one-cycle-per-hop delay found in point-to-point networks. Taken together, these features mean that the ONet is functionally similar to a fully-connected, bi-directional point-to-point network with an additional broadcast capability. Filtering allows the sender and/or the receiver(s) to restrict the set of cores that can receive the message, thereby supporting multicast.

The ATAC architecture was carefully designed taking into account the physical limitations and constraints of both the optical and electronic devices. Based on these constraints, the ONet as described above should scale to at least 64 (and possibly as many as 100) Hubs. This limit is based on several factors including the total optical power a waveguide can carry, the minimum power needed by a photodetector to register a signal, the maximum length of a waveguide, the total range and minimum spacing of wavelengths used for WDM.

These limits can be overcome using multiple waveguides and dividing the communication channels between them. However, eventually the area needed for the optical components will become the limiting factor. The ONet's optical components and photonic interconnect can be placed on a separate layer in the CMOS stack, overlapping the electrical components to which they connect. However, for a 400 mm^2 chip, the entire die would be covered by an ONet with about 384 Hubs. Since chips will eventually grow to thousands of cores, some sharing of Hubs will certainly be needed. Therefore, for the purposes of this paper, we take the simple approach and assume that the ONet is limited to 64 Hubs.

Because of this limit, the set of 1024 cores is broken into 64 clusters of 16 cores that each share an optical Hub. The ONet interconnects the 64 clusters with a 64-bit wide optical waveguide bus. Within a cluster, cores communicate electrically using the EMesh and with the Hub using two networks called the ENet and BNet. The ENet is an electrical mesh that is used only to send data from cores within a cluster to the Hub for transmission on the ONet. The BNet is an electrical broadcast tree used to forward data that the Hub receives from the ONet down to the cores.

Messages from the cores arrive at the Hubs on the ENet. Each Hub then retransmits the data on the ONet using its unique wavelength. Note that this allows the two Hubs shown to send their data simultaneously without interference. The ONet consists of a bundle of waveguides: 64 for data, 1 for backwards flow control, and several for metadata. The metadata waveguides are used to indicate a message type (e.g., memory read, barrier, raw data) or a message tag (for disambiguating multiple messages from the same sender). The receiving Hub captures both of the values simultaneously into sender-Hub-specific FIFOs. These values are then propagated to the cores using the BNet.

IV. Programming Models

ATAC supports multiple programming models that make it easier for programmers to achieve good performance. ATAC's efficient cache-coherence protocols [2] allow a shared-memory model to be used for dynamic or irregular computation. When communication patterns are regular, the ANet enables MPI-style message passing with fast broadcast/multicast for even greater performance. Contrast this with most other architectures where MPI programmers spend considerable effort to avoid using expensive broadcast operations.

While shared-memory and message-passing paradigms are valuable given their widespread familiarity, they don't always give the programmer the right mix of performance and programmability. ATAC's inexpensive broadcast mechanism paves the way towards other programming models that give the programmer an increased level of flexibility to achieve performance near that of hand-tuned message passing with the programming ease of shared memory.

One example of a new programming approach that leverages ATAC's broadcast is Adaptive Constraint-Based Programming

(ACP). Goal-oriented constraint-based problems are becoming increasingly prevalent in various applications domains, including software verification, electronic design automation, general theorem proving, artificial intelligence, and computational biology. Algorithms that solve constraint-based problems are usually built on top of two key engines: SAT solvers for boolean-valued constraints and the Simplex algorithm for real-valued constraints. ATAC enables easy and efficient ACP by leveraging its broadcast network. One approach to find a solution to a satisfiability problem given some constraints is as follows: 1) Broadcast the set of known constraints to all "worker cores"; 2) Each worker core generates a random input vector and, along with the initial constraints, applies traditional SAT solver algorithms to learn new constraints; 3) Worker cores periodically broadcast constraints that they learn to other cores. Given the high efficiency of ATAC's broadcast network, new constraints can be broadcast frequently and with minimal disruption to the senders and receivers. Programming cores to periodically broadcast constraints and integrate new ones makes implementing a high performance version of this algorithm significantly more straightforward than a similar scheme on a traditional electrical mesh network. With a mesh network, achieving high performance would involve carefully orchestrating when constraint exchanging happens (perhaps in a pipelined fashion) to mitigate skew and avoid congesting the network.

ATAC's congestion-free broadcast also enables another form of adaptive, or self-aware, computing: Application Heartbeats [3]. Heartbeats provide a simple programming interface whereby applications publish their performance and system software/hardware can use this information. [3] demonstrates the use of heartbeats in an adaptive H.264 encoder to dynamically reduce output quality or increase computational power to meet a throughput goal. ATAC's broadcast network can be leveraged with such an approach, as worker cores would broadcast their heartrate (in this case, frames per second) to a set of external scheduler cores. If the initial number of worker cores was not able to meet the desired performance expectations, the scheduler cores could dynamically add more cores to the workload. ATAC's optical network allows heartbeats to be efficiently broadcast to all scheduler cores, and the scheduler cores to broadcast new instructions or goals to all worker cores (e.g., "decrease quality").

ATAC also supports a novel programming model called Consumer Tagging, which allow the programmer to specify producer-consumer relationships that are known statically. Consumer Tagging is inspired by Remote Store Programming (RSP) [4]. In the RSP model, standard store instructions are used to transfer data from the registers of a sending processes directly to the local cache of a destination process. Thus, when the consumer reads the data, it will have immediate access to it from its own cache without incurring any coherence overhead. For applications such as H.264 and FFT, Hoffmann et al. showed that RSP can outperform both cache-coherent shared memory and direct memory access (DMA) approaches. More importantly, the RSP approach was as easy to program as shared memory and significantly easier than DMA.

One of RSP's limitations arises when there is more than one consumer for a piece of data. In this case, the producer must perform multiple stores to push the data to all consumers. On a typical electrical network, such stores produce a sequence of individual messages. The costs of issuing multiple stores can significantly impact performance as the number of consumers increases. Hoffmann et al. spent considerable effort minimizing the number of consumers for each piece of data in their H.264 application. Even so, they showed that as the number of cores increased from 2 to 32, the time spent performing multicast RSP operations increased by almost 13×, from 0.3% to 4%. This increase was one of the key reasons why H.264 had the worst speedup of the applications they studied. As multicores continue to grow and applications employ greater parallelism, it will be increasingly difficult for programmers to minimize multicast operations.

Building on the RSP approach, we propose a new programming model for ATAC called *Consumer Tagging*. To use Consumer Tagging, the programmer first writes a regular cache coherent shared memory program. They then profile the application to find performance bottlenecks and modify the program by "tagging" the consumers of data stored at specific addresses. In this approach, the programmer tags an address (or range of addresses) as having one or more consumers. From then on, when the owner core writes a value to an address that has been tagged, the operating system and hardware automatically push a new copy of the data to the list of consumers, all in a way that is transparent to the programmer. When the number of consumers for a given piece of data is greater than one, or if the consumer is known to be far away (say, more than four hops on the electrical network), the underlying system can broadcast the update using ATAC's optical network in one step.

The consumer tagging model makes ATAC easier to program because the application developer can spend less time worrying about overhead and share data in the most natural way. For example, in H.264, the obvious implementation would have resulted in 9 consumers for most shared data but Hoffmann et al. restructured the program to average 2-3 consumers. With the ANet, the cost of communication to N consumers is distributed across the N consumers, so programmers are encouraged to share data liberally. Furthermore, communication latencies are uniform, which frees the programmer from having to carefully optimize the physical locations of communicating cores. Addresses can also be tagged in an optional "full broadcast" mode whereby the programmer doesn't need to specify the list of consumers; updates are simply broadcast to all cores and filtered as appropriate.

In summary, ATAC's inexpensive broadcast offers a number of benefits for applications written using the consumer tagging approach: it reduces the amount of time spent communicating; it allows better overlap of communication and computation because it eliminates a separate communicate phase; and it enables programmers to freely use algorithms with large numbers of sharers, either for performance or ease of implementation. ATAC's optical network also supports an array of other programming models such as cache coherent shared memory, message passing, application heartbeats, and adaptive constraint-based programming—all of which help programmers extract performance out of ATAC with less effort than would be required on traditional multicore architectures.

V. Evaluation

In this section, we evaluate the performance benefits of using the ATAC network. To this end, we compare the ATAC network (*ANet*) to a state-of-the-art pure electrical mesh network (denoted by *pEMesh*) by evaluating the performance of a cache coherent shared memory synthetic benchmark on these networks. The on-chip communication network's workload consists of the cache coherence messages that arise while running this synthetic benchmark. Due to

TABLE I
SYSTEM PARAMETERS

Core Frequency	1 GHz
DRAM Access Time	0.1 μs
Single Hop Latency through Electrical Mesh	1 ns
E/O Conversion + O/E Conversion Time	2.5 ns
Propagation Time though Optical Components	0.5 ns
Link Width of the Optical Broadcast Network (*ONet*)	64 bits
Link Width of the Electrical Networks (*ENet* & *BNet*)	(32 & 64) bits
Link Width of the pure Electrical Mesh (*pEMesh*)	64 bits
Delay through BNet (varies by core)	1 to 3 ns

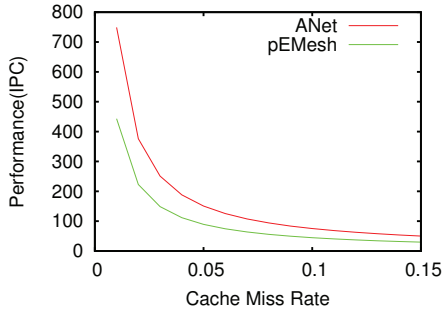

Fig. 3. Performance of *ANet* vs *pEMesh* as a function of Miss Rate

the impracticality of simulating many-core systems such as ATAC with current simulators, we built an analytical model to simulate this benchmark. The model is based on an in-order processor model focusing on latency of memory requests. It takes into account queueing delay in the on-chip network (i.e, at the sending and receiving hubs in ANet and the network switches in pEMesh) as well as off-chip. The analytical model used is described in great detail in [2]. The system parameters used for the evaluation are shown in Table I, mostly being design choices. The optical propagation time is based on the properties of the waveguide materials and the length required to loop through the entire chip.

To evaluate the performance of *ANet* against *pEMesh*, we vary the miss rate of the synthetic benchmark from 1% to 15% and compare the observed instruction throughput. We observe that, on an average, $ANet$ outperforms $pEMesh$ by a factor of 1.69x due to its higher bandwidth, lower latency and broadcast capabilities. To demonstrate this fact, we measure the contribution to the average memory latency due to the following 3 factors: (1) On-Chip Base Latency; (2) On-Chip Queueing Delay; and (3) Off-Chip Latency. Here, the on-chip base latency is the average on-chip latency of a memory request assuming infinite bandwidth.

From our experimental study, we observe that the average memory access time of the synthetic benchmark is 6.26 on *ANet* and 9.26 on *pEMesh*. Out of 6.26 on *ANet*, the contribution of the on-chip base latency is 2.71 (43.3%) and that of the on-chip queueing delay is 0.78 (12.5%). Out of 9.26 on *pEMesh*, the contribution of the on-chip base latency is 5.12 (55.3%) and that of the on-chip queueing delay is 1.37 (14.8%). The contribution due to the off-chip latency is 2.77 in both the cases. From the above figures, we conclude that *ANet* outperforms *pEMesh* in terms of both on-chip bandwidth and base latency. The on-chip base latency is 47.1% lesser in *ANet* as compared to *pEMesh* while the on-chip queueing delay is 43.1% lesser in *ANet* as compared to *pEMesh*. (All latency numbers in this paragraph are reported in terms of processor cycles).

VI. RELATED WORK

CMOS-compatible nanophotonic devices are an emerging technology. Therefore there have only been a few architectures proposed that use them for on-chip communication: Corona [5], the optical cache-coherence bus of Kirman et al [6], and the switched optical NoC of Shacham et al [7].

The Corona architecture primarily differs from ATAC in the way that it assigns communication channels. While Corona assigns a physical channel to each receiver and uses WDM to send multiple bits of a dataword simultaneously, ATAC assigns a physical channel to each sender and uses WDM to carry multiple channels in each waveguide, thereby eliminating contention and the need for arbitration. Kirman et al [6] design a cache-coherent hierarchical opto-electronic bus, consisting of a top-level optical broadcast bus which feeds into small electrical networks connecting groups of cores. The design of their network is similar to ATAC but is limited to snooping cache coherence traffic whereas ATAC is composed of a network supporting a general communication mechanism.

Shacham et al [7] propose a novel hybrid architecture in which they combine a photonic mesh network with electronic control packets. Their scheme is still partially limited by the properties of electrical signal propagation since they use an electronic control network to setup photonic switches in advance of the optical signal transmission. It only becomes efficient when a very large optical payload follows the electrical packet. ATAC, on the other hand, leverages the efficiencies of optical transmission for even a single word packet. Batten et al. [8] take a different approach and use integrated photonics to build a high-performance network that connects cores directly to external DRAM. However, their design does not allow for optical core-to-core communication. An ATAC processor could leverage their design to connect its memory controllers to DRAM.

VII. CONCLUSION

With the recent advances in CMOS-integrated nanophotonics, it is likely that processors will soon incorporate optical components. This paper presented a novel manycore architecture that scales to 1000 cores by embracing this new technology. Results indicate that the ATAC network outperforms a pure electrical mesh network connecting 1024 cores by a factor of 1.68x. This paper also discussed various programming models that are applicable to ATAC, and introduced Consumer Tagging, which leverages unique properties of ATAC to make high-performance parallel programming straightforward.

REFERENCES

[1] J. Psota et al., "ATAC: All-to-All Computing Using On-Chip Optical Interconnects," in *BARC*, 1/2007.
[2] J. Miller, J. Psota, G. Kurian et al., "ATAC: A Manycore Processor with On-Chip Optical Network," MIT, Technical Memo, 2009.
[3] H. Hoffmann et al., "Application Heartbeats for Software Performance and Health," in *PPoPP*, 2010.
[4] ——, "Remote Store Programming: Mechanisms and Performance," in *HiPEAC*, 2010.
[5] D. Vantrease et al, "Corona: System Implications of Emerging Nanophotonic Technology," in *ISCA*, 2008.
[6] N. Kirman et al., "Leveraging Optical Technology in Future Bus-based Chip Multiprocessors," in *MICRO*, 2006.
[7] A. Shacham et al., "Photonic NoC for DMA Communications in Chip Multiprocessors," in *Hot Interconnects*, Aug 2007.
[8] C. Batten et al., "Building manycore processor-to-dram networks with monolithic silicon photonics." in *Hot Interconnects*, Aug 2008, pp. 21–30.