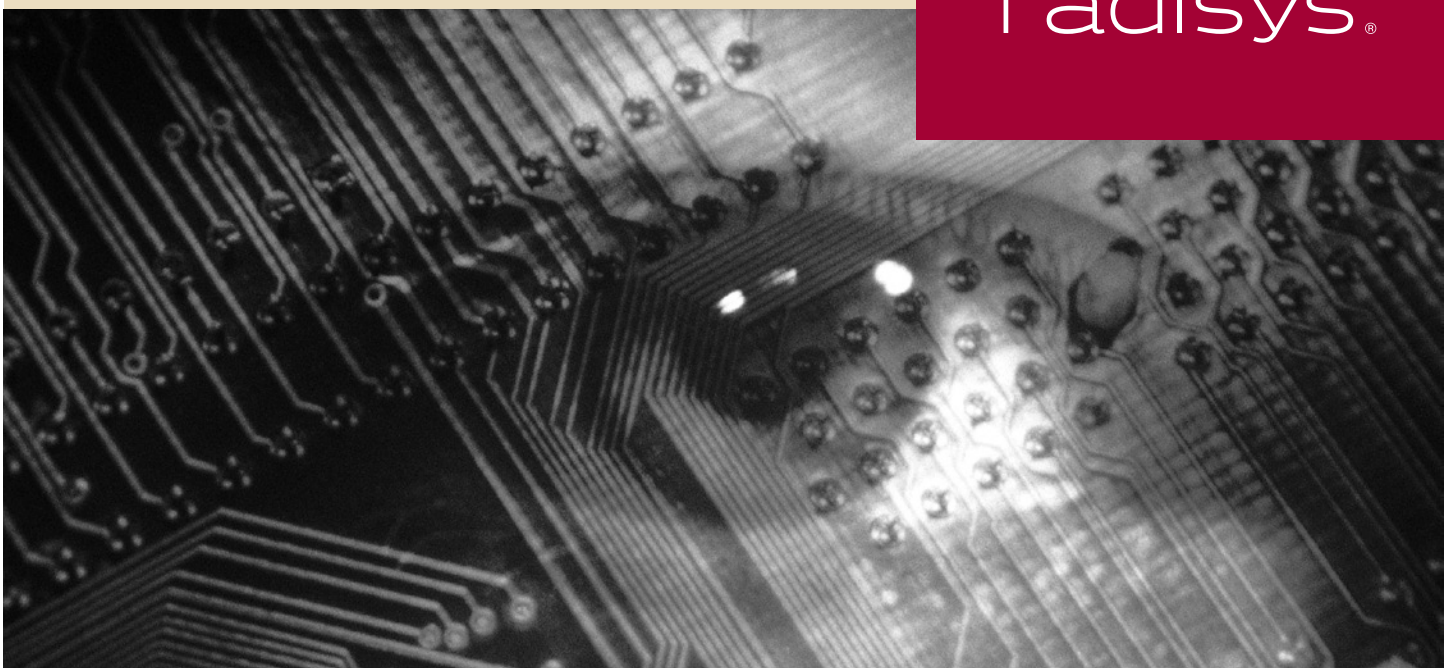
Technology Article

radisys®



Achieving Higher Performance in a Multicore-based Packet Processing Engine Design

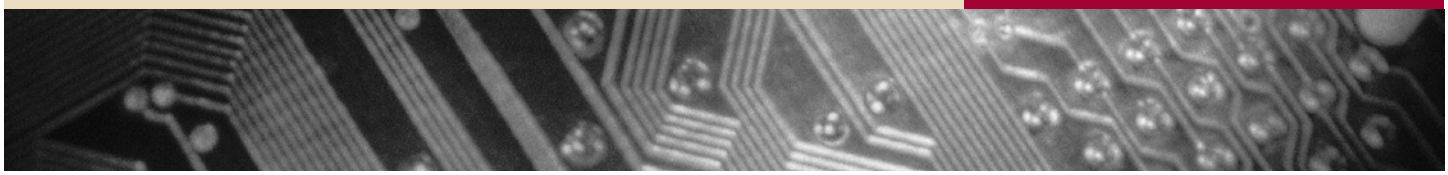
By Mike Coward, VP of Strategy & Innovation

A new class of processor has begun to appear in a variety of storage, security, wireless base stations, and networking applications to replace the very expensive - with long lead times to boot - proprietary Application Specific Integrated Circuits (ASICs) developed by OEM system solution providers as well as those designed by industry giants, such as LSI Logic and IBM.

This class of multi-core processor is made up of eight, sixteen, even sixty-four individual processor cores with integrated memory controllers, various I/O interfaces, and separate acceleration engines.

Though this class of processor has made great strides in overcoming the limitations of earlier generation processors, not all multi-core processors are created equal. Some companies that develop these processors add threading capability to overcome memory latency, and also include native 10Gbps interfaces, while others include security engines and even regular expression engines that support very special applications.

Rather than examining all the features across a number of multi-core processors and comparing them bit by bit, this paper will focus on one critical architectural element, the memory subsystem. The memory subsystem is critical because this is a major factor in determining the scalability and upper limits of performance that a processor can achieve.



The memory architectures compared here are based on two leading multi-core processors in the market today:

- Single channel, wide cache line (Single / Wide)
- Dual channel, narrow cache line (Dual / Narrow)

The question to be addressed is: Which architecture is superior in providing the performance necessary to keep up with the ever growing voice, video, and data traffic that the market is requiring today?

Single Channel, Wide Cache Line (Single / Wide)

The single channel, wide cache line approach uses a single memory channel as the interface between the processor and DDR2 memory. The width of the channel is 128-bits and uses 16-bits of ECC for a total of 144-bits. In this "Single / Wide" approach, cache lines of 128-bytes are used and every access to memory is a burst-of-8 reads or writes.

The result of this approach is that every burst to memory fills or empties a single cache line. With support for DDR2-800 memory, the Single / Wide approach has a memory bandwidth of 12.8GBps, and is achieved by supporting a potential of 100 million transactions per second, where a transaction is either a read or a write of a 128-byte cache line.

Dual Channel, Narrow Cache Line (Dual / Narrow)

The dual channel, narrow cache line architecture uses a different approach for maximizing memory performance. The "Dual / Narrow" architecture utilizes two memory channels as the interface between the processor and DDR2 memory where each channel is 64-bits wide with 8-bits of ECC. The cache lines in this architecture are 32-bytes and every access to memory is a burst-of-4 reads or writes. This architecture similarly fills or empties an entire cache line with a single transaction. The Dual / Narrow architecture achieves the same 12.8GBps raw memory bandwidth, but reaches this figure through 400 million possible transactions per second.

From a theoretical perspective, at DDR2-667 speeds, the Single / Wide memory interface performance is 83 million cache line operations per second, while the Dual / Narrow approach is 334 million cache line operations per second. However, DDR2 memory is far from ideal and has a number of factors that reduce the theoretical performance, including:

- Refresh times
- Bus turnaround times
- Bank access time limitations

Simulations were developed to compare the two architectural approaches. For a typical configuration of 4GB of DDR2-667 memory and a packet classification workload as described below, the Single / Wide architecture yields 64 million cache line operations per second, while the Dual / Narrow architecture yields 204 million cache line operations per second.

It is important to note that although the Single / Wide architecture has an efficiency of 77%, [64MOps actual / 83MOps potential], compared to 61% efficiency [204MOps actual / 334MOps potential], the Dual / Narrow architecture provides more than three times the number of transactions per second. As discussed below, this plays a significant role in packet throughput in real applications.

A Common Application - Load Balancing / Packet Distribution

AdvancedTCA (ATCA) packet processor blades are often called upon to act as a front-end for an entire chassis of blades. In these applications, the packet processor connects to the network on one side and to a set of application blades on the other side.

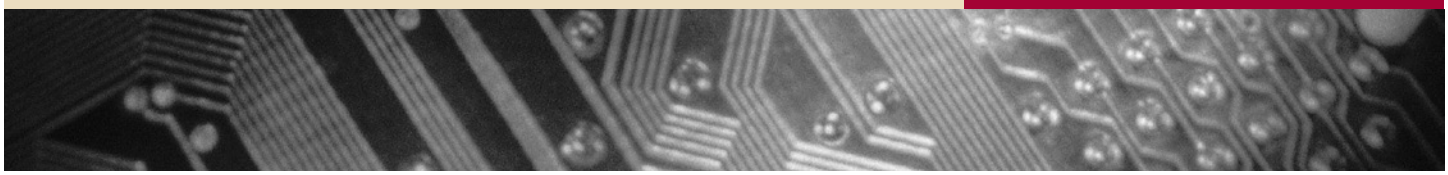
Furthermore, the packet processor blade acts as load balancer and allows the entire collection of application blades to appear as a single IP address, critical to hide the internal complexities of the system from the network.

To gain an understanding for the challenge a solution must undertake to perform 10Gbps of load balancing and network address translation (NAT), consider a system specified to run at 10Gbps with minimum sized 64-byte packets, which is 16.4 million packets per second, in each direction, or 32.9 million packets per second through the packet processor.

An optimized load balancer / NAT engine will execute the following steps for each packet:

- Receive packet and place into cache memory
- Perform a flow lookup
- Modify the packet header per the flow
- Increment statistics about the packet / flow
- Send the packet from cache to the next process

Note that this represents the best case - the packet is never stored to DRAM - only to cache memory, so the number of memory accesses is kept to a minimum.



Flow Lookup Algorithms

As packets are received into the system, they must be categorized as to whether or not they match an existing flow or are part of a new flow. This is normally done using a 5-tuple match, where the five fields that define the flow are matched against a database of existing flows:

- Source IP Address
- Source Port
- Destination IP Address
- Destination Port
- Protocol

The most common lookup function to check a database of existing flows is a hash lookup. Hash lookup is where a key is created based on the 5-tuples and then indexed into a list of matching keys.

The keys point to records that define each flow and records may be chained together in case multiple 5-tuples hash to the same value. Each lookup requires a minimum of two memory lookups, one to search the list of keys and a second to retrieve the flow record. If multiple flows hash to the same key, additional memory accesses will be required to follow the list of chained records.

In order to minimize the number of collisions, the number of hash buckets is normally chosen to be at least 2x larger than the number of expected flows, and even with 2x buckets, 2.24 memory accesses will be required on average. With 10x more buckets than flows, this drops to 2.05 memory accesses per packet.

Statistics. Once the flow has been located, statistics about the flow must be updated. In the highest performing NAT engines, these statistics are stored in the same cache line as the flow record, meaning that the statistics are already in memory once the flow has been located. Once the statistics are incremented, the cache line must be written back to main memory, requiring one further memory access.

Cache Performance. These flow lookups and statistics update operations make the cache memory perform poorly because the number of packet flows tends to be much larger than the number of cache lines, meaning that a given flow is unlikely to be in main cache at any given time.

Example: Assume 500K flows, with 4M hash buckets. If each hash bucket is an 8-byte pointer, and each flow record is 32-bytes, then the hash table is 32MB (4M * 8-bytes), and the flow table is 16MB (500K * 32 bytes). With a 2MB cache, the chance that a given flow will already be in cache is only 4% (2 / 48). With the 3.05 memory accesses required per packet, the cache only has a small impact and drops the average memory accesses per packet to 2.93.

Required Memory Performance

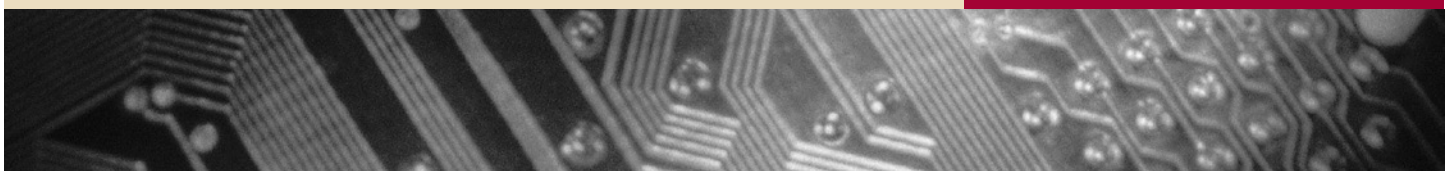
A highly optimized load-balancing engine / NAT engine can be created requiring on average 2.93 memory accesses per packet. Given the memory throughput for the Single / Wide and Dual / Narrow architectures discussed previously, the maximum packet rate and throughput for the two architectures can be calculated as shown in Table 1 below.

This table highlights the impact of the memory architecture differences between the Single / Wide and Dual / Narrow approaches. The Single / Wide approach is only at 66% of line rate with DDR2-667 and cannot reach 10G full-duplex even with DDR2-800 memory.

On the other hand, the Dual / Narrow architecture easily reaches 10G even with the slowest DDR2-400 memory, and with standard DDR2-667 memory the architecture delivers more than twice the memory performance required for full duplex 10GbE; thus, providing significant headroom for additional lookups and advanced functions.

NAT / LB Function (2.93 memory accesses/packet)			NAT / LB % of Full Duplex Ethernet w. 64-byte packets	
Memory Speed	Single / Wide Architecture Mpps	Dual / Narrow Architecture Mpps	Single / Wide Architecture %	Dual / Narrow Architecture %
DDR2-400	13.7	46.8	41%	142%
DDR2-533	17.7	57.7	54%	175%
DDR2-667	21.8	69.6	66%	212%
DDR2-800	25.3	75.1	77%	228%

Table 1. *Comparison of memory architectures*



The reason for the large difference between the two architectures can be found in the cache line differences. The Single / Wide approach is designed with unusually large 128-byte cache lines, but typical network and packet processing applications require only 8- and 32-byte lookups.

As a result, most of each cache line is wasted. The Dual / Narrow architecture, on the other hand, has a cache line size of 32-bytes which more closely matches what is required in typical network and packet processing applications and results in higher performance.

Memory Access Budget. A second way to look at the problem is to calculate the number of DDR memory accesses allowed per packet at 10G full-duplex. With 32.9 million packets per second, the Single / Wide architecture allows 1.9 DDR memory accesses per packet, while the Dual / Narrow architecture permits 6 DDR memory access per packet. Again, the Dual / Narrow architecture provides much higher performance.

Summary

When evaluated against a simple load balancing / NAT application, even when highly optimized to require less than 3 memory accesses per packet, the Single / Wide approach cannot deliver 10Gb line rate full duplex performance, while the Dual / Narrow architecture provides twice the necessary lookup bandwidth.

Most packet processing applications are considerably more complex than this simple load balancer / NAT application and do require more lookups and statistics updates.

In addition, this analysis did not include any overhead for slow-path processing, fast-path management, or security processing, which suggests that the true performance of the Single / Wide approach will be even lower than analyzed here. Ultimately, the Dual / Narrow architecture is required to achieve 10Gbps line rates and above in network and packet processing applications.

radisys®

Corporate Headquarters
5435 NE Dawson Creek Drive
Hillsboro, OR 97124 USA
503-615-1100 | Fax 503-615-1121
Toll-Free 800-950-0044
www.radisys.com | info@radisys.com

February 2010