Data Plane Packet Processing on Embedded Intel® Architecture Platforms

November 2009
Executive Summary

Data plane packet processing involves moving data from an I/O device to system memory, classifying the data and then moving the data to a destination I/O device as quickly as possible. At the high speeds of modern communication, this puts pressure on the system bus as data is moved between I/O devices and system memory, and the processors classifying the data. This application is made even more challenging under a distributed memory architecture, where minimal and deterministic I/O latency must be ensured. This paper describes techniques that can be used to overcome these technical challenges and achieve high-performance data plane packet processing on embedded Intel® architecture platforms.

# Contents

Under the Hood ......................................................................................................................... 4
- Key Platform Features ........................................................................................................... 4

The Application .......................................................................................................................... 5
- Features of the Packet Processing Application ....................................................................... 6

Application Software Architectures .......................................................................................... 7
- System Software ...................................................................................................................... 8

Application Optimization ......................................................................................................... 8
- Multi-core Processing Optimization ........................................................................................ 9
- Memory Optimization .............................................................................................................. 10
- Key Memory Optimization Techniques ................................................................................... 10
- NUMA-aware I/O Optimization .............................................................................................. 10

Looking Ahead .......................................................................................................................... 11
- Future Intel® Xeon® Processor Codename “Westmere-EP” ...................................................... 11
- Future Intel® Xeon® Processor Codename “Jasper Forest” ....................................................... 11
- Key Benefits of Processor Codename “Jasper Forest” ............................................................ 12

Summary .................................................................................................................................... 13
Under the Hood

Embedded Intel® architecture platforms offer high-performance processing machines that comprise the latest Intel® Xeon® processor with four cores per processor. Each processor supports Simultaneous Multithreading Technology (SMT) in a Dual Processing (DP) configuration interconnect with the I/O Hub (IOH) chipset through the latest Intel® QuickPath Interconnect (Intel® QPI) system bus offering high-bandwidth data transfer between the processors and the IOH chipset. Besides, each processor supports three channels of DDR3 memory providing a high throughput for memory accesses to its memory subsystem.

Key Platform Features

- Intel® Xeon® processor 5500 series operating at 2.53 GHz core frequency in Dual Processing Topology with integrated DDR3 memory channels populated with 2 GB, 2Rx8 DIMMs on each channel over 6 channels (at 144 GB maximum with Dual Rank DIMMs).
- Intel® QuickPath Interconnect running at 5.86 GTs.
- Intel® 5520 Series IOH Chipset supporting 40 lanes of PCI Express* (PCIe*) Generation 2 with speed at 5 GHz (see Figure 1).
- Intel® ICH10 Chipset supporting multiple I/O interface.
- Four Intel® PRO/1000 PT Quad Server Adapter cards with Intel® I/O Acceleration Technology add-in card support.
Figure 1. Two Processor Sockets Platform Using the Intel® 5520 Chipset System Block Diagram

The Application

A packet processing application running on this platform delivers high throughput network processing optimized using a number of hardware acceleration features. These hardware acceleration features are part of the enhancement made in the Intel® Xeon® processor based on the Intel® microarchitecture, codename "Nehalem". The core supports more streaming instructions for doing more within one core clock cycle and the processor also supports integrated memory channels providing higher bandwidth for system memory accesses.

Additionally, the processor supports a higher throughput system bus architecture using Intel® QuickPath Interconnect and support higher bandwidth of PCI Express* Generation 2 for I/O devices.
Packets received from the Network Interface Cards (NICs) are classified into flows, where each flow has an associated action that is executed once a packet has been classified. The typical actions that are performed on a packet are either simple forwarding or Network Address Translation (NAT). Flow processing begins with the receipt of a packet from the NIC. The kernel space NIC driver is notified of the packets arrival by the operating system. In this case, four Intel® PRO/1000 PT Quad Port Server Adaptor NICs are used to offer the full sixteen Gigabit Ethernet ports of traffic.

After a packet is received by the application, the classify function is called to determine the flow to which the packet belongs. This lookup is based on 5-tuple classification values from the packet header: Source Address, Destination Address, Source Port, Destination Port and Packet Type. If the flow for this packet is found, the Action Handler is called to transmit the packet from the correct port. Otherwise, the packet goes through a Policy Lookup function to establish a flow for this packet type (5-tuple). Once the flow is established, the packet is handled by the Action Handler function and is transmitted to the appropriate port. The Action Handler function supports packet filtering, decision making for passing the packet or dropping the packet and also NAT functionality for header modification.

**Features of the Packet Processing Application**

- One kernel thread per interface.
- MSI interrupt (IRQ Affinity) for each interface tied to a specific core.
- New API (NAPI) mode soft interrupt fires; notifies the kernel that there is data, which then polls until empty.
- 256K entry table per interface, up to 4 million flow entry hash lookup table size.
- Exception flow implemented as another kernel thread.

### Application Software Architectures

#### Figure 3. Linux* Kernel Software Stacks in the NIC Routing Packet Process

**Figure 3** shows the software stacks in the Linux* kernel that are involved in routing the packets to and from the NICs. The left side of the figure shows the unmodified version of the Linux network stack, while the right side of the figure shows the modified e1000e driver for the Network Card and the Packet Processing Application used.

1. The Network Card triggers an interrupt to indicate that new packets have arrived.
2. The Linux* kernel calls the e1000e interrupt handling function, which notifies the NAPI layer of the interrupt and returns.
3. NAPI in turns calls the e1000e poll routine to start processing the received packets.
4. In the case of the Linux Network Stack, the received packets are passed up the stack for processing.
5. In the case of the Packet Processing Application, the application processes the packets and makes the decision to forward them to another port or drop them.
Figure 4 shows the results of benchmarking packet processing with the Intel® Xeon® processor E5540, operating at a core frequency of 2.53 GHz in a Dual Processing configuration. The test environment was configured to achieve optimal throughput.

The system demonstrates the ability to sustain 17 million packets per second for the shortest Ethernet packets (64 bytes). This is the worst case performance for the system to keep up with the arrival rate of these short packets. As the packet size grows, the percentage CPU utilization reduces.

**System Software**

- CentOS v5.2 x64 bit
- Vanilla Linux® kernel 2.6.27.23
- Packet Processing Application

**Application Optimization**

The Packet Processing Application has its own lockless packet receipt and transmission mechanism, independent of the Linux kernel. The mechanism is designed to scale well in an environment with multiple threads, cores and flows. The stack that a packet must traverse between receipt and transmission has been significantly reduced in size versus the Linux kernel.
network stack. This implies that fewer clock cycles are consumed by each packet, leading to improved throughput. In addition, memory alignment and cache coherency of the application have been improved to further optimize data movement within the system.

Packet processing can be further optimized by disabling the flow management functionality that deals with flow aging. This means that once a flow is established, it never expires. Also, flows are cached to further improve the flow lookup process instead of reading from memory.

**Multi-core Processing Optimization**

The Intel® Xeon® processor on a DP platform offers a total of 16 logical processing units to handle a wide load of network packets from multiple network ports in the system. *Figure 5* shows the affinity between packet processing threads and the execution core of the processor socket.

*Figure 5. Packet Processing Threads and Execution Core Diagram*

This affinity reduces remote cache accesses by tying Ethernet threads to a unique core, thus keeping the data locally in the processor's cache, reducing
缓存插槽间的处理器缓存 thrashing 通过 Intel® QPI 接口。远程缓存访问可以进一步减少，通过配置流量，使得将数据传输到和接收的以太网端口，可以绑定到相同的核心上的对称线程。

**Memory Optimization**

Intel® Xeon® 处理器 5500 系列平台提供了一种分布式的内存拓扑结构，包括本地和远程内存，以及缓存结构和系统内存。该平台提供了一种高效的缓存一致性实现，利用了高带宽和高性能的 Intel® QuickPath Interconnect。该系统。

**Key Memory Optimization Techniques**

- **Lockless Queuing**: 这种方法消除了保护数据队列的需要。在多线程环境中，这种方法提高了效率，消除了线程间对资源竞争的需要和对同步的需要。
- **Cache Alignment**: 这种方法将数据缓冲器对齐到缓存行边界，减少了部分缓存行更新，进一步提高了缓存的效率。
- **Memory Address “swizzling”**: 内存地址“swizzling”将数据缓冲器对齐到所有可用的 DDR3 DIMMs 在各种内存通道中的最佳使用。最有效的内存使用模型是跨内存通道和 DDR3 DIMMs 交错请求，避免返回到同一内存 DIMM，从而避免预充电 penalty。在这样，内存读取和写入在 DIMMS 之间平衡，避免了单一 DIMM 或通道的瓶颈。
- **Descriptor Bunching**: 这种方法通过将更新合并到传输环的尾指针来减少 PCI 配置空间的写入。结果，这种技术减少了将更新写入 PCI 配置空间的数量。

**Note**: 内存对齐和内存地址“swizzling”是互斥的。

**NUMA-aware I/O Optimization**

该网络接口卡系统驱动程序进一步修改以减少远程内存访问。这些会导致应用的延迟。Packet Processing Application 允许对内存分配进行精细粒度的控制，使得为每个数据处理线程分配的内存来自本地到核心的内存池和执行该处理的线程。
Looking Ahead

Future Intel® Xeon® Processor Codename “Westmere-EP”

Intel is currently developing a socket- and pin-compatible processor code name “Westmere-EP” that will provide migration and upgrade from a total of eight cores with Intel® Xeon® processor 5500 series to a total of 12 cores per system (Westmere-EP) in a two processor sockets solution (see Figure 6).

This future processor offers an upgrade solution to existing platform solutions offering more processing headroom for adding functionality and upgrade to the existing solution space.

Figure 6. “Westmere” Two-processor Socket Platform Diagram

Future Intel® Xeon® Processor Codename “Jasper Forest”

Meanwhile, Intel is also developing a more integrated system solution with an upcoming processor code name “Jasper Forest”. This processor offers an
integration of the existing Intel® Xeon® processor 5500 series with sixteen PCI Express® Generation 2.0 lanes (see Figure 7). Also, this processor offers additional Non Transparent Bridge (NTB) functionality in a port of the PCI Express® support and other RAID acceleration features.

This future processor offers more system integration and provides for more dense embedded system solutions.

**Figure 7. “Jasper Forest” Two-socket Processor Platform**

**Key Benefits of Processor Codename “Jasper Forest”**

- Reduces total system BOM cost
- Reduces system solution form factor
- Reduces power consumption
- Increases system integration level
- Increases ease of debug
- Increases usage model with support of Non Transparent Bridge on PCI Express® port
Summary

This paper showcases embedded Intel® architecture platforms with multi-core processors as an excellent vehicle for delivering high-performance data plane packet processing to meet today's data plane processing challenges.

The Intel® Embedded Design Center provides qualified developers with web-based access to technical resources. Access Intel Confidential design materials, step-by step guidance, application reference solutions, training, Intel’s tool loaner program, and connect with an e-help desk and the embedded community. Design Fast. Design Smart. Get started today.
Data Plane Packet Processing on Embedded Intel® Architecture Platforms

Authors

Ai Bee Lim is a Senior Technical Marketing Engineer with ECG at Intel Corporation.

Ray Kinsella is a Network Software Engineer with ECG at Intel Corporation.

Acronyms

IOH  I/O Hub
NAT  Network Address Translation
NIC  Network Interface Cards
SMT  Simultaneous Multithreading Technology