Keywords: multi-core processor, multi-thread

Abstract: This document introduces the advantages of multi-core multi-thread processors in IP networks and the highlights of the multi-core multi-thread processors adopted by HP A6600 routers.

Acronyms:

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Full spelling</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIC</td>
<td>Application Specific Integrated Circuits</td>
</tr>
<tr>
<td>NP</td>
<td>Network Processor</td>
</tr>
<tr>
<td>ERP</td>
<td>Enterprise Resource Planning</td>
</tr>
<tr>
<td>CRM</td>
<td>Customer Relationship Management</td>
</tr>
<tr>
<td>SCM</td>
<td>Supply Chain Management</td>
</tr>
</tbody>
</table>
Table of Contents

Background .......................................................................................................................... 3
Evolution of the IP Network ............................................................................................... 3
Evolution of the Router ....................................................................................................... 3
Bottlenecks of Traditional Single-Core CPU ................................................................. 4
Challenges of Network Processor .................................................................................... 5
Emergence of Multi-Core Processor .................................................................................. 5
Advantages of Multi-Core Processor .................................................................................. 7
Features of A6600 Multi-Core Multi-Thread Processor .................................................. 9
  Multi-Core processing service in parallel ................................................................. 10
  Multi-Thread Processor .............................................................................................. 11
  Inner Communication Bus of multi-core processor ................................................... 12
  Embedded Hardware Accelerator .............................................................................. 12
    Load Balancing Engine ............................................................................................ 12
    Crypto Engine ........................................................................................................ 12
Summary ............................................................................................................................ 13
Background

Since its emergence, the Internet has become indispensable to daily lives, from working, traveling, shopping, to any other aspect of lives. Ever-growing network users and their demands for more diversified services have impelled the development of network devices to carry more services and provide ample throughput. Underpinning this development is the development of chip technology.

Evolution of the IP Network

The development of IP networks usually can be divided into three phases:

- **Phase I (before 2001):** IP networks only provide simple connections. At this phase, there were few service types and critical services. IP networks mainly carried data services such as Email, file transfer, and information release. The carried traffic was mostly text and pictures. Because the networks transmitted traffic in the best-effort mode without any access control for services or users, differentiated services could not be achieved.
- **Phase II (between 2001 and 2003):** IP networks provided access control and user management. At this phase, IP networks broke away from the extensive development mode, and an operable and manageable network model was advocated. This model won great success in both technologies and marketing. Even though operable and manageable technology somewhat intelligentized the edge networks, prominent issues such as quality of service, security, availability, and service management were still there.
- **Phase III (2003 to present):** IP networks have developed with the aim of providing high-quality services. With the fast development of IP network technologies and services, IP networks have carried more types of services including data, voice, and video and thus more critical businesses such as ERP, CRM, SCM, electronic business, production, and finance services. Different services are different in requirements for QoS, security, and availability. Therefore, the IP networks must be capable of providing differentiated services.

QoS, security, availability, service performance and scalability, and service management are now important for an IP network. The development of fundamental carrier networks has manifested the integration of IP packet switching and connectionless technology with the design idea of telecommunication networks. This is the result of an effort to build a larger, faster, secure, more reliable, and flexible manageable networks to carry various emerging applications.

Evolution of the Router

Since the emergence of the Internet, routers, being the core processing devices for IP network, experienced several technology innovations in the past several decades. The evolution of router can be divided into five phases:

- **Phase 1:** Centralized forwarding, single-core CPU as the engine, shared bus, fixed interfaces
- **Phase 2:** Centralized forwarding, single-core CPU as the engine, shared bus, modular interfaces
- **Phase 3:** Distributed forwarding, single-core CPU as the engine of line-card, modular interfaces
- **Phase 4:** Distributed forwarding, ASIC as the engine of line-card, modular interfaces
Phase 5: Distributed forwarding, NP as the engine of line-card, modular interfaces

As we can see, the evolution of routers is essentially a process from centralized architecture to distributed architecture, from single-core CPU to ASIC and NP (programmable ASIC) for service processing, and from fixed interfaces to modular interfaces. These five phases also roughly reflect the technology evolution of routers.

How will routers evolve along with the development of network services? What opportunities and challenges will network devices, especially routers, encounter in functionality and architecture?

So far, incorporating new generation multi-core processors that feature simple programming, easy upgrading, and massive throughput in routers have become a trend in the industry.

**Bottlenecks of Traditional Single-Core CPU**

The performance improvement of single-core CPU mainly depends on the improvement of processor frequency. Due to production technology restrictions, single-core CPU has encountered performance ceiling in both performance/price ratio and performance/power ratio.

**Figure 1** Bottlenecks of single-core CPU

Moreover, the processing performance of a processor also depends on the rate of memory and I/O peripherals accessing. As shown in the diagram above, the development of the processor frequency, the memory access rate, and the I/O access rate is unbalanced. The processor frequency is doubled every two years, the memory access rate is doubled every six years, and the I/O access rate is doubled every eight years. The imbalance has brought about severe bottlenecks for performance improvement. It is infeasible to improve system performance by only improving processor frequency because the CPU has to wait for the response from the memory or I/O for a long time in order to continue processing, which is unbearable for high-speed network service processing. Also, high frequency processors require extremely sophisticated production technology. Production difficulty and low rate of finished products thus bring ever-lasting high production costs.

Finally, power consumption and performance/power ratio must be taken into consideration during hardware system design to reduce power consumption and heat dissipation. Else it will add risk to the availability of the system.
Challenges of Network Processor

Network processors (NPs) are expected to provide flexible programmability while delivering performance like ASICs. However, limited microcode space and inflexible service processing procedure prevent NP-based routers from responding to customer requirements rapidly in a service oriented network, which is critical to business success. Additionally, designing and maintaining NP-based routers require rich experiences and great efforts because of its technical difficulties in microcode programming and performance optimization.

From the perspective of service processing, traditional embedded CPU (like MIPS) or general purpose CPU (like X86) can process and respond to higher-layer services more easily and rapidly. Besides, it is not difficult to find experienced developers for a CPU-based system.

Thus, many new devices, features, and concepts are first implemented on CPU platforms, then fixed via ASIC to improve performance. This shows that CPU is a good choice of verification platform for processing new services. With the cost factor excluded, neither ASIC nor NP will be needed if CPU can deliver satisfactory performance for service processing.

Emergence of Multi-Core Processor

From the perspective of router vendors, an ideal processor dedicated for network communication should feature simple programming, easy upgrading, and massive throughput.

Figure 2 Microprocessor competition situations

As shown in the diagram above, microprocessors used for network devices can be divided into four categories: embedded CPU, general purpose CPU, ASIC chip, and NP (Network Processor).
In terms of forwarding capability, their order is ASIC, NP, embedded CPU, and general CPU from the highest to the lowest.

In terms of Layer-4 through Layer-7 services processing flexibility, their order is general purpose CPU, embedded CPU, NP, and ASIC from the highest to the lowest.

The following describes these four types of processors in detail.

- **General purpose CPU**

  General purpose CPUs refer to x86 series CPUs. The major vendors are Intel and AMD. It usually has a 2 to 3 GHz working frequency and hyper pipeline design. The strengths of general purpose CPUs include the most flexible programmability, the best adaptation to L4-L7 services, the simplest application development environment, and the largest developer resource pool.

  General purpose CPUs are not dedicated communication processors. However, as networking applications become more and more flexible and complicated, general purpose CPUs sometimes are used in network products. Unfortunately, without dedicated high-speed communication buses and lack of hardware acceleration for packet processing in IP networks, the overall packet forwarding performance is not good. Thus, general purpose CPU is not a good choice for line-card. However, considering the computation capability of general purpose CPU is very powerful, some vendors adopts it as the core of MPU to do routing calculation for better convergence time.

- **Embedded CPU**

  Embedded CPUs, such as ARM, MIPS, and PPC chips, are widely used in routers. Most routers in Phase 1, 2, and 3 are based on embedded CPUs as mentioned.

  The processor frequency of an embedded CPU is in the range of hundreds of MHz to 2 GHz. Even though this is not very high, the embedded CPU demonstrates higher forwarding performance than a general purpose CPU because its system architecture is optimized for packet processing.

  Moreover, embedded CPUs can process L3-L4 services and even L4-L7 services. However, due to computation capability restriction and lack of dedicated hardware acceleration components, embedded CPUs' performance is poor when handling complicated services. In addition, embedded CPUs are not as good as general purpose CPUs in service flexibility.

- **Dedicated ASIC chips**

  Dedicated ASIC chips were introduced to satisfy the explosive increase in requirements for network bandwidth. Because IP forwarding and MAC forwarding are fixed in ASIC chips, the ASIC-based packet forwarding performance can easily reach multi-10G level, which is much higher than any embedded or general purpose CPU. Therefore, dedicated ASIC chips are widely used in Ethernet switches and routers in phase 4 as mentioned. But it also creates an embarrassing situation: the service processing capability is not commensurate with the high-performance bandwidth offered to users through ASIC and thus additional devices must be used to provide required service processing capability.

  For example, to allow LAN users to access the Internet via NAT, an ASIC based Ethernet switch needs to be coupled with a standalone NAT gateway or NAT service blade. As for complicated network applications such as encryption and voice, current ASIC chips can do few things. Since supported applications are almost fixed in ASIC chips, you need new generation ASIC chips to
process new services. Customer need to purchase new devices, which is clearly not a cost-effective solution.

- NPs

NPs were introduced to satisfy requirements for high-performance services in routers. NPs can be regarded as programmable ASICs. Though NPs’ packet forwarding performance is a bit lower than ASICs, NPs can process L3-L4 services that ASIC cannot process because NPs are programmable. For example, NPs can process NAT, GRE tunneling, and L2TP tunneling with high performance. However, as NPs are developed in microcode, it takes a long time to develop a new function on NPs. Restricted to limited microcode space, an NP cannot support too many services. Restricted by hardware architecture, NPs cannot process complicated services such as tunnel encryption and multi-services overlapping.

As the result of an effort to integrating the benefits of variety of processors, the multi-core processor dedicated for network communication (different from X86 multi-core CPU) was introduced, intended to deliver high performance, easy programmability, and flexible adaptation to L4-L7 services. So far, this is the best approach to chip performance improvement under the current power consumption restriction.

A multi-core processor contains multiple independent physical cores. Each core can be a center for computation, system management, data forwarding and service processing. Each core has an independent logical structure, including the buffer memory, execution unit, instruction level unit, and bus interface. Cores communicate with each other via high-speed bus and shared memory. In practice, each core can even work in a power saving mode. Though the calculation speed of a single core decreases, the total performance can be multiplied through the cooperation of remaining cores.

### Advantages of Multi-Core Processor

**Table 1 Processors at a glance**

<table>
<thead>
<tr>
<th>Feature</th>
<th>Single-core CPU</th>
<th>NP</th>
<th>Multi-core Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Programmability/ease-of-use</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Running OS</td>
<td>Supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>C/C++ language, standard instruction set</td>
<td>Supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Unlimited instruction space</td>
<td>Supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Memory protection</td>
<td>Supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Packet forwarding performance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Optimized packet forwarding instructions</td>
<td>Not supported</td>
<td>Supported, microcode-based</td>
<td>Supported</td>
</tr>
<tr>
<td>High-efficient memory sub-system</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Optimized packet re-assembly, forwarding, buffering, and scheduling</td>
<td>Not supported</td>
<td>Supported</td>
<td>Supported</td>
</tr>
<tr>
<td>Feature</td>
<td>Single-core CPU</td>
<td>NP</td>
<td>Multi-core Processor</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>--------------------------</td>
<td>-------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>High-capacity data cache</td>
<td>Incompletely supported</td>
<td>Not supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>

### Memory and security processing acceleration

<table>
<thead>
<tr>
<th>Feature</th>
<th>Single-core CPU</th>
<th>NP</th>
<th>Multi-core Processor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hardware encryption</td>
<td>Incompletely supported, limited</td>
<td>Not supported generally</td>
<td>Supported generally</td>
</tr>
<tr>
<td>Regular expression, pattern adaptation</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Incompletely supported</td>
</tr>
<tr>
<td>TCP hardware acceleration</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Incompletely supported</td>
</tr>
<tr>
<td>Hardware compression and decompression</td>
<td>Not supported</td>
<td>Not supported</td>
<td>Incompletely supported</td>
</tr>
</tbody>
</table>

As shown in the table above, multi-core processors deliver these benefits:

- Easy programmability and short development period
- Powerful data forwarding performance
- Embedded hardware acceleration for complicated services

They are described in the following subsections in detail.

**Easy programmability and short development period**

Both multi-core processor and single-core CPU can be programmed in advanced languages like C, can run OSs, and have unlimited instruction space. You can use the OS to manage hardware and software resources to simplify programming and facilitate inheriting and implanting existing codes. As a result, multi-core processor and single-core CPU are best in programmability and ease-of-use.

In contrast, NPs cannot run OS and can use only proprietary development environments provided by the vendors and are programmed by microcode. Though you can program in C language on some vendors’ NPs, the performance and efficiency will be greatly reduced. NPs’ instruction code space is also limited, so NPs are deficient in programmability and ease-of-use. As a result, NP-based products respond and adapt to new services more slowly than CPU-based products. Each NP vendor has some restrictions on the service processing procedure and system architecture, which makes NPs hard to respond to diverse L4-L7 services flexibly.

**Powerful data forwarding performance**

Similar to NPs, multi-core processor supports optimized packet forwarding instruction and highly efficient memory sub-system, so they can process and forward data packets more rapidly; multi-core processor is optimized particularly in packet reassembly, forwarding, buffering, and scheduling; multi-core processor has large-capacity multi-layer data cache. As a result, multi-core processor’s performance in packet forwarding and packet processing is very high.

Comparatively, single-core CPUs are not well optimized in data packet forwarding, plus linear processing mode, therefore they cannot satisfy the ever-increasing requirements for forwarding performance of network devices.
Embedded hardware acceleration for complicated services

In the system architecture design phase, multi-core processor focuses on built-in hardware acceleration for complicated services, such as application recognition and encryption services urgently needed in the market. Usually, a multi-core processor is accelerated by powerful embedded hardware engines in processing, so that the multi-core processor can save many processor resources from strenuous tasks, such as encryption and decryption, compression, traffic regular expression check etc. Thus the performance of multi-core-based service processing can be well assured.

Features of A6600 Multi-Core Multi-Thread Processor

HP A6600 routers (referred to as A6600 in this document) are multi-service aggregation routers developed for carrier and enterprise networks. A6600 adopts industry leading multi-core multi-thread processor as data forwarding and service processing engine. Therefore, it has flexible service customization capability like CPUs and also guarantees unparalleled service processing performance.

Figure 3 Architecture of A6600 multi-core processor

As shown in Figure 3, A6600 multi-core multi-thread processor adopts system on a chip (SoC) technology, which integrates network high-speed bus like 10G duplex SPI4.2, Load Balancing Engine (LB) and crypto-engine on a chip, adopts 90nm CMOS production technology, and is integrated with more 300 million transistors.

A6600 multi-core processor has eight cores (Core1 through Core8), and each core has four hardware threads as indicated in yellow ellipses. Each thread has its own independent register groups and does not need context switching during thread scheduling, so that the 32 threads can all run at high speed in parallel. Currently, A6600 multi-core multi-thread processor supports main stream embedded operation systems like Real-time Linux and VXWORKS etc, which make the development quite easy.
The multi-core multi-thread processor has many highlights. The following part introduces some of them in the aspects of multi-core, multi-thread, inner communication bus, and hardware accelerator.

**Multi-Core processing service in parallel**

*Figure 4* Service parallel processing model of a multi-core processor

Traditional CPU processes all services by one single core. As a result, the workload of the CPU is very heavy, and services may interrupt due to interferences with each other. For example, if the packet forwarding workload is too heavy, the local service processing, such as routing packet transceiving and calculation, will be affected, and even routing interruption may occur.

Multi-core processor can solve this problem perfectly.

As shown in *Figure 4*, different cores of A6600 multi-core processor can process services in parallel, such as firewall, encryption, Netstream traffic analyzing, QoS scheduling, and control plane services in parallel. Additionally, load balancing can be implemented across cores. Consequently, the services will not interfere with each other, and the system’s availability and service processing performance are both greatly improved.
Multi-Thread Processor

Figure 5 multi-thread improves the processing efficiency

From the perspective of inner cores, it is generally considered in the industry that multi-threading is an effective solution for memory access and I/O access delay. The Intel P4 hyper-threading (HT) technology is a successful application of two threads. A P4 HT CPU can run multiple threads to process more instructions and data. Hyper-threading fully utilizes temporary idle resources in the processor. Similarly, each core in an A6600 multi-core processor has four hardware threads, totally 32 hardware threads, and thread scheduling is completed by hardware at high efficiency.

A multi-thread processor enables the system to do more things during the unavoidable delay for accessing memory or I/O. As shown in Figure 5, when four tasks identified by different colors are processed within a single-thread processor, the processor must read memory before processing each task and then continues to process the task after the read operation returns. As the memory is slower than the processor, it takes a long time to access memory. As shown in the diagram, it takes T1+T2 to finish four tasks with 1-thread processor. In case that the four tasks are processed in a 4-thread processor, when the first task is waiting for the return from the memory accessing, the processor can continue to process the second task with another hardware thread. The processing of the third and fourth task is similar. In this way, the processor still processes tasks during the memory accessing delay, and therefore the efficiency of processing multiple tasks is greatly improved. As shown in the figure above, the time of processing the same four tasks in a 4-thread processor is reduced to T1. A6600 adopts 32-thread processor, so that it can better utilize the system resources and enjoys incomparable advantages in processing multiple tasks without any doubt.
Inner Communication Bus of multi-core processor

Inner communication bus design affects the performance of processor greatly, especially for multi-core multi-thread processor. Inside a multi-core multi-thread processor, the cores, the threads, and the peripheral components must collaborate with each other, so that the efficiency of communication them is extremely important for the system service processing performance. As shown in Figure 3, A6600 multi-core processor adopts the Fast-Message-Network (FMN) technology to connect all the cores, threads, multiple network interfaces, Direct-Memory-Access (DMA), and security engines at a high rate.

FMN is a message network with a bandwidth of primary frequency × 64 bits. For instance, if the primary frequency is 1 GHz, the FMN bandwidth is 64 Gbps, and the FMN almost can be considered as an infinite-bandwidth and non-blocking network. Thus FMN enables messages to be exchanged among key components inside the processor at a high rate simultaneously. With FMN, the arbitration phase for common buses is also avoided, so the inner communication efficiency of the processor is greatly improved.

Embedded Hardware Accelerator

It is hard to optimize the system processing performance only with programming techniques. On the other hand, programming techniques cannot improve the system processing performance to a large extent. Currently, hardware accelerator is commonly used in the industry to improve performance. A multi-core processor is usually embedded with some hardware accelerators to assist the cores in processing services. HP A6600 multi-core multi-thread processor is embedded with the following two powerful hardware accelerators, which do not occupy system resources when working.

- load balancing engine
- crypto engine

Load Balancing Engine

Load balancing is the basis for multi-core processing. Without load balancing, some cores may carry too much load, while some cores may be idle, and therefore, the system performance improvement is restricted.

A6600 multi-core multi-thread processor is embedded with a powerful load balancing engine (LB engine) as shown in Figure 3. The LB engine can rapidly pre-process the contents of data packets entering the chip through the network interface. For example, if packets enter the chip from an Ethernet interface, the engine can resolve the Layer-2, Layer-3, and Layer-4 information of Ethernet packets flexibly based on multiple policies (which are programmable), extract character strings, complete packet check and authentication automatically, and calculate the hash value. Then based on the hash results, LB engine distributes these packets to the dedicated cores for further processing. In the whole procedure, load balancing is implemented without the participation of CPU cores, and the processing capability of each core is fully utilized. As a result, the service processing performance is greatly improved.

Crypto Engine

With the popularization of networks, more and more services are carried by IP networks and various network attacks and information interception are rampant throughout networks. Thus network
application security draws more and more attention. One solution for network security is to use IPSec
to encrypt packets. But if IPSec service is processed with software, the performance is unacceptable; Purchasing dedicated IPSec module or service blade will increase investments a lot.

HP A6600 routers adopt multi-core multi-thread processor embedded with high-speed crypto engine supporting industry standard encryption algorithms such as AES, DES/3DES, SHA-1, SHA-256, and MD5 etc. This engine helps A6600 achieve industry-leading encryption performance without purchasing any dedicated encryption cards. As proofed by the authoritative certification authority Tolly Group, IPSec throughput of one line-card reaches 3.8Gbps, with 400 IPSec tunnels configured at the same time.

Summary

It is obvious that multi-core architecture becomes the trend of processor development. And new generation multi-core multi-thread processor, processing Layer-4 through Layer-7 services in parallel, is the ideal choice for service aggregation router design.