

# Intel Based Architecture Vs ASIC Based Architecture

---

## Technical Brief

### **Overview**

The document explains the merits of Intel and ASIC based architectures. In the broader context we will explore the architecture benefits as applied to Application Delivery Controllers, which offer a multitude of functions.

## Introduction

The document explains the merits of Intel and ASIC based architectures. As for which is better, neither is really any better than the other. Both offer superior benefits if applied and used in the right context when building enterprise class networking products. The networking vendors that choose a particular architecture over the other have implications on the devices performance, scalability and reliability. In the broader context we will explore the architecture benefits as applied to Application Delivery Controllers, which offer a multitude of functions.

## ASIC Architecture

### Benefits and Issues

#### Benefits

ASIC's are intended to perform a single well defined function to provide scalability and performance. The key word is well defined requirements. The blind store and forward mechanism of the switch (which is a non data-intensive operation) enabled vendors to choose ASIC architecture. Many ADC vendors started out building devices based on Intel architecture and later integrated ASIC's to perform some customized well defined functions like SSL offloading and compression (FPGA's). The same vendors started using ASIC's to offer switching functions in an ADC.

#### *Switching*

Classic switching operates at layer2 (MAC layer) of the TCP/IP stack in the networking model. The switch needs to blindly forward packets just by reading the control information of the IP packet without touching the data portion of the packet. This requires the switch to do a basic parsing of the control information for packet forwarding which is not very process intensive. The function is very clearly defined and ASIC is the best solution as it is tailored to perform a single customized function in the most efficient manner. As such switches could offer enterprise class performance by adopting ASIC as the basic framework.

### *SSL Offloading*

SSL transactions chew up CPU cycles of an ADC because of the expensive data intensive packet processing involved in encryption and decryption of the packets on a repeat basis. As SSL offloading involves repeated encryption and decryption, an ASIC architecture brings the needed scalability to process voluminous transactions.

### *Compression*

Compression is used to reduce transmission bandwidth and decrease the latency for faster response. Compressed data must be decompressed to be understood. The repeated compression and decompression of the data consumes intensive CPU cycles. ADC is flexible in choosing a compression algorithm that is appropriate based on the nature of the application, nature of the client and data patterns. The repeatability and flexibility requirements of compression in ADC can be achieved using CPU's or via special purpose ASIC or FPGA hardware. Flexibility requirements are easily met with a hardware FPGA solution than an ASIC due to ASIC's set in stone design and well defined requirements constraints.

## **Issues**

### *HTTP and Applications*

ADC's have evolved from just manipulating TCP and HTTP protocol headers to manipulating the data portion of the packets. Data portion can contain application specific control or data information that needs to be understood by the ADC. The constant changing nature of application requirements, the complexity of the HTTP 1.0/1.1 protocol and specific application protocols (related to CRM, ERP and e-Commerce) requires that an ASIC be rebuilt frequently to match with the speed of changing application requirements from customers. Integration of applications with HTTP poses huge challenges as applications use HTTP protocol in a non-standard way. The design of an ASIC becomes unmanageable and complex restricting its use in an ADC to certain well defined protocols and functions.

## *General Issues*

ASICs once designed are cast in stone; bug fixes are extremely costly, since they need a new spin of the ASIC depending on the fix. Additional functionality is impossible to add when customers need new enhancements in a timely manner. Traditionally the cost of an ASIC based device has always been expensive due to the long manufacturing, engineering and testing cycles.

## *ASIC and ADC*

Switch manufacturers tried moving up the TCP/IP stack (primarily TCP, SSL and Compression) to offer complex Application Delivery Controller functions like server load balancing, proxy, compression, encryption and security in a single device, which requires huge amount of data-intensive processing on a per packet basis. The non data-intensive design of the switch is no match for the data-intensive ADC due to the heavy duty packet processing requirements. The complexity of ADC's requirements eventually forced the ASIC based switch vendors to integrate high performance server processors from AMD or Intel into their switch based architectures to meet the performance, scalability and reliability requirements. Truthfully, ASIC vendors rely on general purpose CPU's for ADC's core functions like Layer 7 load balancing and application delivery.

## **Intel Based Server Architecture**

### **Benefits and Issues**

#### **Benefits**

Intel's initial goal was to build a generic enough processor that can perform myriad of functions. So the design was loosely coupled with the function. In the recent times, Intel started to offer several processor lines focused on the server market. The Pentium and the Xeon (Vs AMD Opteron) server line of CPU's have been deployed by many of the enterprises to perform data-intensive operations. Similar to a server, Application Delivery Controller (ADC) needs to perform lot of data-intensive processing on a per-packet (HTTP, SSL or TCP protocol based) basis to achieve the desired functions on the higher layers of the TCP/IP stack (primarily Layer 7 HTTP and SSL). The Intel server processors design was inline with the requirements of ADC vendors as they provided superior performance and scalability while performing data-intensive

operations in base software OS. The processors provided performance, scalability and reliability that is required of an ADC. The above mentioned reasons made the Intel based architecture a natural choice for ADC vendors and they started out building devices based on Intel architecture. Later, ADC vendors integrated ASIC's to perform some customized functions like SSL offloading, compression (FPGA's) and lastly switching.

### *Caching*

A proxy simply terminates connections but a cache has to understand HTTP protocol in-depth in order to make caching decisions. For a cache this is an extremely data intensive operation as the TCP/IP stack has to de-capsulate and encapsulate packets at multiple layers and reassemble them in the process. In essence cache's need to understand HTTP protocol completely. Caching is very demanding and expensive operation compared to server load balancing or switching. Caching involves fast parsing of the application data for a quick lookup in the cache data store of an ADC. Depending on the nature and content of the application data streams, parsing can consume a wide range of CPU cycles in an ADC. Besides HTTP protocol is not as well defined as TCP or IP protocol. Device vendors exploit the flexible rules of HTTP protocol and implement non-standard functions to satisfy functional and performance requirements of specific applications. Intel architecture based CPU's are well designed in managing diverse software requirements of application caches in an ADC.

### *Server Load Balancing*

Server load balancing (SLB) operates in the layer 4 and layer 7 of the TCP/IP stack. The server load balancer device needs to understand both the data (with varying degree of granularity) portion and control portion of the packet to make advanced application load balancing decisions. Layer 4 SLB operates at the TCP layer which is more or less well defined and is easier to implement in an ASIC. Layer 7 traffic is rich with application data that is very unique based on the application. This requires the underlying software processing architecture to be extremely flexible to understand unique applications to make intelligent load balancing decisions. AMD or Intel based server CPU architecture fits the flexible heavy duty processing requirement of the Layer 7 SLB. The reality is that most ASIC based ADC vendors perform layer 7 SLB (layer 4 SLB in some cases) functions on an in-built Intel or AMD server processors.

## Issues

### *Switching*

Switches blindly forward millions of packets all day long without the need to perform data-intensive processing on a per-packet basis. They just need to look at the control portion of the packet and forward them accordingly. Due to the simplicity of the switch behavior the functions were well defined. General purpose processors which are flexible did not provide the added performance and tuning needed for scaling switch functions as they are not designed to perform a well defined function efficiently.

### General Issues

The general purpose processors limitation is based on Moore's Law, which states that the processor speeds will double every 18 to 24 months. Intel is able to achieve these targets although beating the speed is perceived as a challenge in the coming decade by the pundits. Intel made dramatic break through from 90 nm to 45 nm chip designs and doubled the speeds in addition to adding multiple cores. ADC's have to depend on a third party processor manufacturer for product delivery.

## Summary

Array Networks APV employs a best of breed approach to provide maximum performance, scalability and availability. APV employs a hybrid architecture that combines ASIC's, Intel CPU's and FPGA's tailored to meet the heavy duty demands of an Application Delivery Controller (ADC), while still offering best performance, quality, feature functionality, scalability and ROI.