SOLUTION BRIEF
AUGUST 2017

AMD EPYC

HIGHLIGHTS

- ○ MULTI-CHIP MODULE FOR ENHANCED SCALABILITY
- ○ NUMA ARCHITECTURE FOR HIGH PARALLELISM
- ○ HIGH CORE COUNT
- ○ LARGE MEMORY CAPACITY
- ○ LOTS OF I/O BANDWIDTH
- ○ FAST INFINITY FABRIC INTERCONNECT
- ○ HIGH MEMORY AND I/O BANDWIDTH TO REDUCE LOADED LATENCY
- ○ HIGH CAPACITY FOR DENSE CONSOLIDATION AND LESS CLUSTER COMMUNICATION
- ○ LOW CACHE LATENCY FOR FAST RESPONSE TIMES
- ○ FAST, PRIVATE L3 CACHES TO SUPPORT TYPICAL VIRTUAL MACHINE SIZES

Multi-Chip Module Architecture: The Right Approach for Evolving Workloads

The latency story isn't what some suggest. With a forward-looking multi-chip module (MCM) approach, the AMD EPYC™ SoC delivers the scalability and performance applications need to deliver business outcomes.

Recently AMD announced AMD EPYC, a system on chip (SoC) that delivers real innovation to better support the needs of existing and future datacenter applications. Using an innovative multi-chip module approach, the AMD EPYC SoC can help IT organizations balance what applications need and what IT infrastructure delivers. This document shares our vision for the AMD EPYC SoC architecture and explains the importance of memory latency and how it affects system scalability and performance.

THE NEED FOR A NEW APPROACH

The x86 architecture has seen only incremental improvements over the last several years. Indeed, the last three significant innovations were developed during an era of intense market competition between multiple vendors. Sixty-four-bit addressing, multicore processors, hardware-accelerated virtualization—all of these innovations were developed during an era of healthy competition in the x86-architecture CPU market and all first developed by AMD.

As the automatic leaps in processor performance once predicted by Moore's Law become increasingly elusive, innovation is becoming even more important today. New and evolving applications demand more and faster bandwidth, quick access to large data sets, and accelerated parallel processing. As the server marketplace enters a new epoch, gains in software can finally accelerate with better support from the underlying silicon—if it has the right design.

THE MOVE FROM SINGLE DIE TO MULTI-CHIP MODULES

Traditional processors place one or more CPU cores on a single die to accelerate clock rate and cache access. That's a reasonable approach for some CPU-intensive processes, but it has limitations. Oxide thickness and transistor length, width, and density can only be pushed so far before gates "leak" and too much heat is generated. In short, the physics

Multi-Chip Module Architecture

of a single-die approach simply don't scale to support the number of processor cores that today's applications need.

Understanding that servers need to keep pace with applications, we designed the EPYC SoC using a multi-chip module (MCM) approach. The MCM architecture uses several dies, each comprised of multiple cores and memory stacks.

As information flows through the system, it's important for inter-chip communication to be fast and latency kept to a minimum. In the AMD EPYC SoC, dies are interconnected using a fast, coherent AMD® Infinity Fabric with speeds up to 170 GB/second. The innovation and tremendous scalability enabled by Infinity Fabric and a non-uniform memory architecture (NUMA) played a major part in our ultimate decision to implement an MCM solution. This enterprise-grade fabric helps the AMD EPYC SoC deliver the scalability needed by most of the server market today, and we believe that it will continue to be the right architecture to address the server market over the next several years.

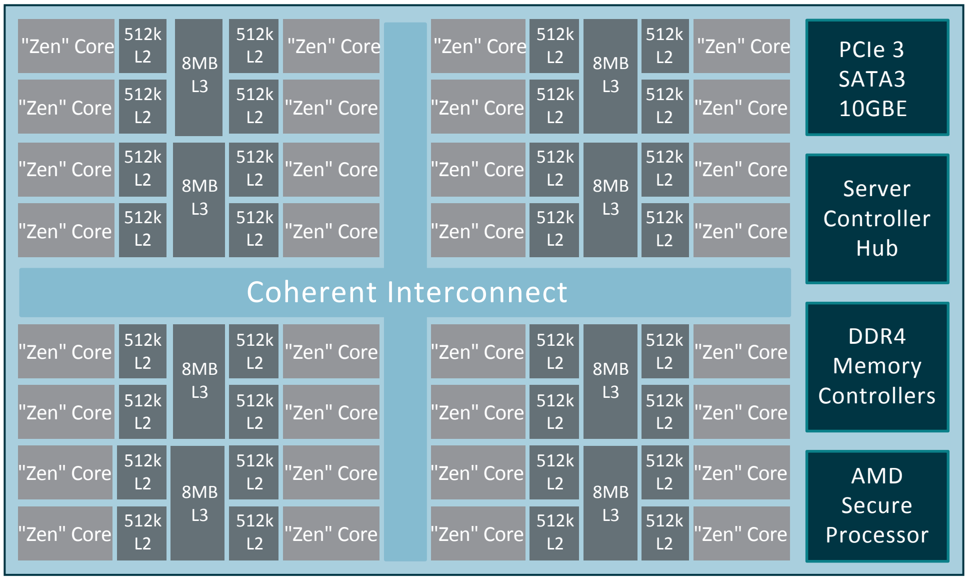
THE BENEFITS OF NUMA ARCHITECTURES

Architected for scalability, NUMA architectures have been at work in server and processor designs for decades. Indeed, some of the largest servers in terms of parallel processing capabilities were designed in the late 1990s—the Sun Enterprise 10K server and the HP Superdome, for example. All used NUMA architectures with operating system support for locating data near the processors that use it. Those servers were designed with modular capacity where each unit of scale incorporated a small set of processors with memory. The set of processors in the module had lowest latency to the local memory, and higher latency to memory attached to other modules. These servers were the workhorses for enterprise applications at the end of the last century, and their performance scaled very well as resources were added. Handling non-uniform memory access was a well understood operating system concept at that time and each of the competitors mentioned (Sun, HP) had NUMA solutions.

NUMA EVOLUTION

The drive for more parallelism and processing power in 2- and 4-socket CPU configurations moved us to a model where the concepts once used at the server level were then used between individual CPUs. Indeed, AMD introduced NUMA architectures at the CPU level with AMD Opteron™ processors and HyperTransport in 2003, followed by Intel in 2009. These products exhibited one latency for local memory access and another for memory accessed through the "other" CPU.

Today, the drive for more parallelism and processing power in a single CPU propels us to using a NUMA model with a multi-



Multi-Chip Module Architecture

chip module. This model uses the same architecture for scalability used in large servers, multi-socket servers, and now within a single SoC.

Servers built with the AMD EPYC SoC have memory latency that varies depending on where the data is that a particular core needs to access. Operating systems and hypervisors are built to create an affinity between software and data, locating data in the memory closest to the core that will use it. AMD works with major vendors to help them optimize their software to be aware of the AMD EPYC implementation of NUMA.

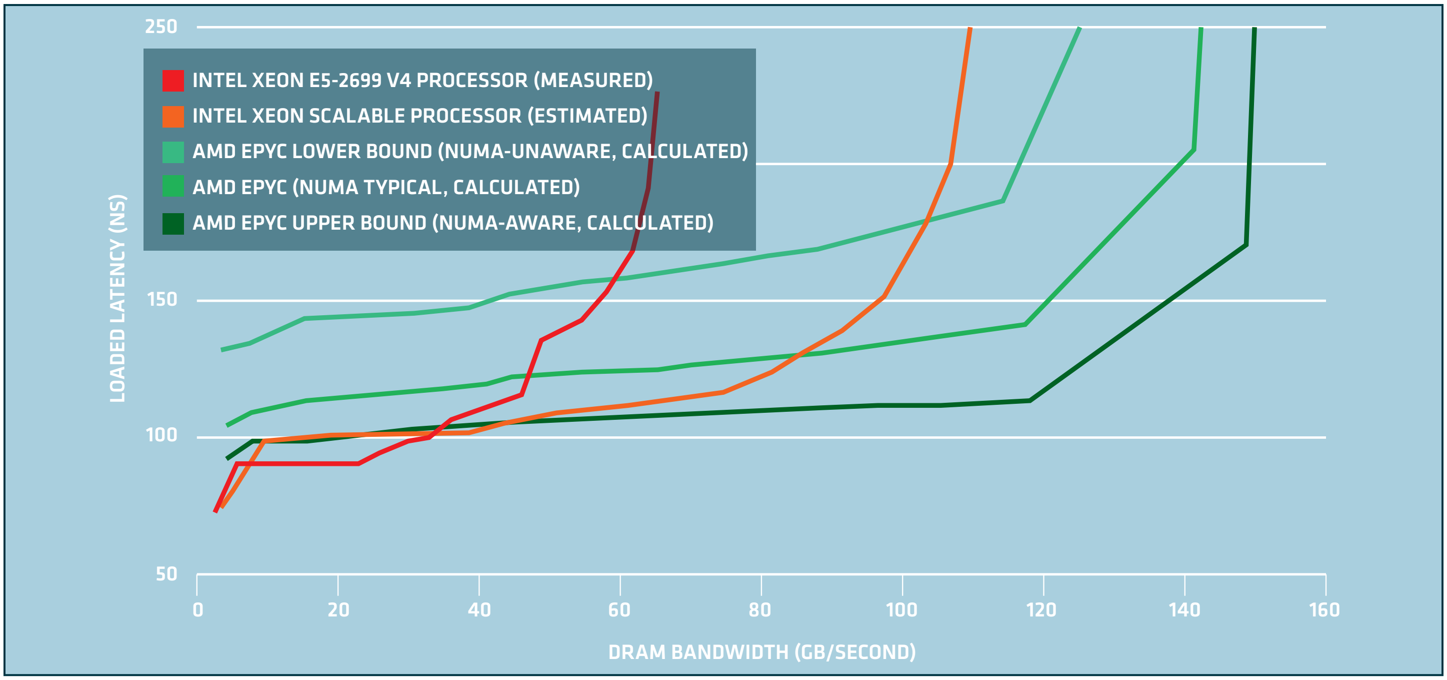
NUMA offers an architecture for scalability but also more flexibility to optimize the latency and bandwidth trade-off that best suits your needs. If your software is highly latency sensitive, you can create multiple small NUMA domains that tightly couple regions of memory to a CPU die. If your software is bandwidth sensitive, you can create a single large NUMA domain and interleave allocated memory locations so that all of the SoC's memory channels are evenly utilized.

MEMORY LATENCY MYTHS

In NUMA architectures, it's easy to compare minimum (for the closest memory) and maximum (for the most distant memory) latencies, but this "unloaded" latency just defines the range of possible latencies. Operating systems, hypervisors, and applications optimize memory placement and processor affinity so that workloads exhibit memory access times that are closer to the lower end of the spectrum. Whether software is NUMA-aware or not, it can benefit from other AMD EPYC SoC features that help reduce loaded latency effects.

A RANGE OF DRAM LATENCIES

We designed the AMD EPYC processor with twice the memory channels of the Intel Xeon processor E5 v4 family (8 versus 4) and one-third more memory channels than Intel Xeon Scalable processors (8 versus 6). This



AMD testing conducted by AMD Performance Labs on an AMD Grandstand reference platform configured with 2 x EPYC 7601 processors fixed at 2.2 GHz, 16 x 32GB DDR4 2666MHz DIMMs, Ethanol BIOS version "TSW1000C" and Ubuntu Server 16.04.2 LTS-kernel 4.10.0-22. Using AMD internal loaded latency test and estimations with 1 latency thread per die , and 7 load threads per die (1 thread per core). Loaded latencies are based on targeting all threads to their local memory access (NUMA Aware), all threads uniformly across remote memory accesses (NUMA Unware), or a weighted average of these latencies (Typical).

Multi-Chip Module Architecture

modularity supports today's highly evolved NUMA ecosystem. And it allows our real-world performance to enjoy greater headroom before the limit of memory bandwidth is reached.

PERFORMANCE UNDER LOAD

As workloads execute transactions and make more intensive use of cores, memory, and I/O, latency increases slowly. When limits are reached, latency then increases dramatically. This "loaded" latency is an interesting aspect to analyze because it exemplifies what real-world applications experience on a daily basis.

AMD EPYC has the cores, memory, and I/O to help enterprise workloads perform. For example, AMD EPYC can help:

○ CLOUD DATACENTERS SCALE AND PERFORM. Support for the cloud is baked in the silicon, making an AMD EPYC SoC-based system ready to support public, private, and hybrid clouds. More cores enhance parallelism enabling greater virtual machine density, and the capability to assign virtual machines to cores can improve the capability to meet service levels. More I/O capacity supports the massive amounts of east-west bandwidth needed for communication between virtual machines. More memory capacity enables more robustly configured virtual machines to be hosted, a benefit to VDI environments as well.

○ HPC APPLICATIONS CAN RUN FASTER. More cores means more workload components can execute on a single server, allowing data to be exchanged between components at memory, not interconnect speeds.

○ MACHINE LEARNING APPLICATIONS SUPPORT DECISION-MAKING ON A REAL-TIME STREAM OF INCOMING DATA. More memory and I/O bandwidth support faster loading of data streams from external sources, and directly from disks with built-in SATA controllers. And more PCIe® lanes means a larger number of high-performance NVMe devices can be directly accessed for data caching. Compare this to processors with fewer I/O channels in which latency-inducing PCIe switches must be employed in order to provide access to more devices.

○ ENHANCE THE USER EXPERIENCE IN VDI ENVIRONMENTS. Single-socket server configurations based on AMD EPYC SoCs increase the ratio of graphics accelerators to CPU cores, contributing to a better user experience. A large number of cores and I/O channels supports the connection of additional graphic accelerators that can be connected to enable low-latency graphics rendering. Compare this to processors with fewer PCIe I/O lanes in which latency-inducing PCIe switches must be added to accommodate more graphics devices.

LOTS OF BANDWIDTH

Loaded latency is a function of bandwidth. The longer it takes to transfer data into and among system components, the longer the CPU must wait, slowing application response.

○ I/O BANDWIDTH. AMD EPYC has the I/O bandwidth to support the movement of data to and from the network, spinning disks, NVMe storage, and graphics acceleration devices. Indeed, the highest number of PCIe® lanes[3] in the industry helps meet the east-west bandwidth needs of all environments, including cloud computing and big data clusters.

○ NUMA DOMAINS AND INFINITY FABRIC. In a typical shared-memory processor, contention results when two or more CPUs try to address the same memory on a shared bus. The NUMA architecture used with the AMD EPYC SoC gives each processor access to its own local memory (a NUMA domain) but also allows each processor to access memory owned by another processor across a high-speed Infinity Fabric. The AMD EPYC SoC supports up to 8 NUMA domains, providing scalability and reducing the contention of CPUs competing for access to memory across a shared bus.

○ FAST CACHES. One way to accelerate access is to use high-capacity, low-latency caches. Why? Many application requests are fulfilled from cache. The AMD EPYC SoC uses 512KB L2 cache per core (16 MB total L2 cache) and 8 MB shared L3 cache per 4 cores (64MB total L3 cache). These cache sizes are optimized to balance latency and cost anticipating where the server market is headed, with typical deployments using from one to four cores per virtual machine.

Multi-Chip Module Architecture

LEARN MORE

○ TO LEARN MORE ABOUT THE AMD EPYC SoC, VISIT AMD.COM/EPYC.

○ MORE MEMORY AND MORE MEMORY BANDWIDTH. More processing power and I/O capacity makes software hungry for more memory and more memory bandwidth. Both of these help move applications and their data in and out of the CPU more quickly. With the highest amount of memory[1] and bandwidth[2] per x86-architecture CPU in the industry, the AMD EPYC SoC can accommodate more virtual machines per CPU and speed their performance with larger memory footprints. That means you can increase the capacity of in-memory databases, speed big data analysis, and cache more incoming data from the Internet of Things. And you can speed your enterprise applications with this optimal balance of memory capacity.

MOVE FORWARD WITH AMD

No matter what you need to do—host virtual machines, deploy virtual desktop infrastructure (VDI) or software-defined storage infrastructure, or run GPU-accelerated high-performance computing (HPC) applications, data analytics, or other workloads—AMD EPYC SoCs can help you optimize the scalability and performance of your workloads.

As with any system, your performance may vary from what industry-standard benchmarks are able to reveal. For example, benchmarks that exercise one aspect of CPU performance may not show how your workloads will actually run on an AMD EPYC SoC-based server. The best approach is to carefully assess how your workload is affected by memory latency by giving your real-world workloads a test drive on AMD EPYC SoC-based servers.

DISCLOSURES

1. AMD EPYC offers up to 128GB LRDIMM in 2 DIMM per channel configuration, so up to 256GB/channel x 8 channels = 2.048 TB/processor, versus the Xeon E5-2699A v4 processor at 128GB LRDIMM in 3 DIMM per channel configuration, so up to 384GB/channel x 4 channels = 1.54 TB/processor. NAP-04

2. AMD EPYC supports up to 21.3 GB/s per channel with DDR4-2667 x 8 channels (total 170.7 GB/s), versus the Xeon E5-2699A v4 processor at 19.2 GB/s with max DDR4-2400 x 4 channels (total 76.8 GB/s). NAP-03

3. AMD EPYC offers up to 128 PCI Express high speed I/O lanes per socket, versus the Xeon E5-2699A v4 processor at 40 lanes per socket. NAP-05