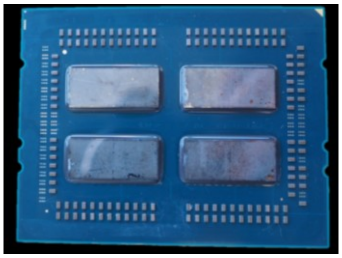
Introduction

Designing a server chip is an exercise in balance, juggling between system cost, die area, memory bandwidth, memory latency, *etc*. Five years ago, as AMD weighed re-entering the server market, it had a difficult task ahead of it. The company needed a processor that had performance competitive with Intel's Xeon processors, yet could offer differentiation, all while still maintaining 100% x86 software compatibility. What AMD accomplished with its EPYC processor appears to be just that. The company designed and manufactured a server processor that offers more memory and I/O bandwidth than Intel's Xeon designs, yet is also less expensive to build than if AMD had built a large monolithic die. AMD achieved that goal by using the efficiencies of multichip module (MCM) technology and the company's new Infinity Fabric (IF) technology.

The MCM technology (Figure 1) offered AMD a chance to build a highly capable server processor using smaller and more manufacturable die. AMD also developed the Infinity Fabric, an extension of the company's HyperTransport technology, as a seamless, scalable interconnect that could be used for on-die, on-package, and multi-package communications. The resulting design includes a hierarchical path between its constituent processor die, leading to a non-uniform memory access (NUMA) model.

AMD is well versed in NUMA architecture design. When AMD first introduced the Opteron processor in 2003, its new HyperTransport interconnect enabled a direct-connect NUMA architecture between Opteron sockets, scalable from two to eight processor sockets. NUMA enabled the sole Opteron single-threaded core in each socket to access memory across the interconnected sockets.

Figure 1: EPYC Multi-Chip Module Design

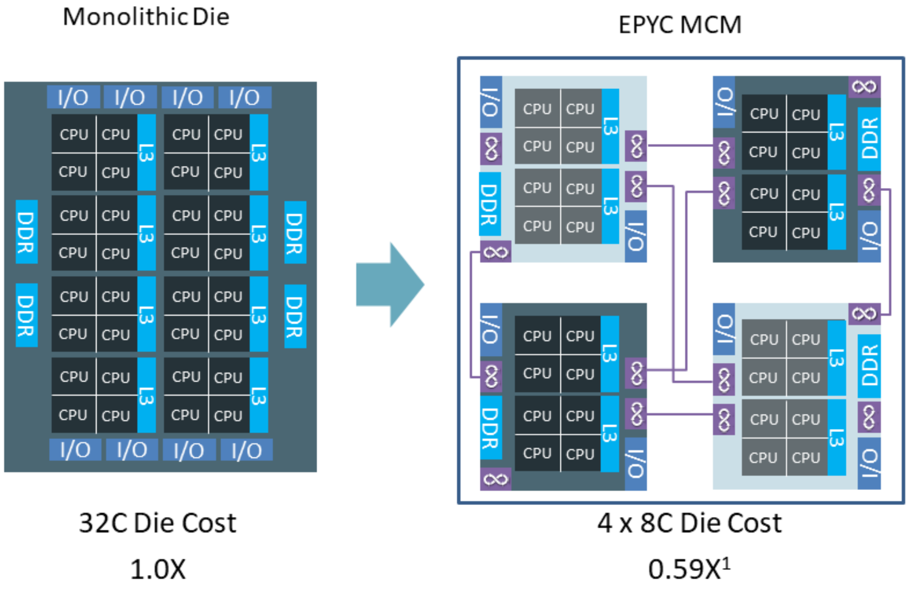


Source: TIRIAS Research

In 2005 AMD introduced the dual core Opteron, which enabled NUMA for four to 16 single-threaded cores across the memory attached to those two to eight sockets. With the EPYC processor, AMD has extended support for up to 32 dual-threaded cores in a single package (for 64 hardware threads per socket), all NUMA enabled. Plus, AMD supports dual-socket EPYC system designs with up to 64 cores and 128 threads, likewise NUMA enabled.

The base EPYC building block die has 8 cores and was code-named "Zeppelin". By combining four Zeppelin die in one package, AMD can deliver a 32-core processor but with better economic and cost structure than if it was a giant monolithic 32-core die (Figure 2).

Figure 2: EPYC Multi-Chip Yield Tradeoff



1. Based on AMD internal yield model using historical defect density data for mature technologies.

Source: AMD

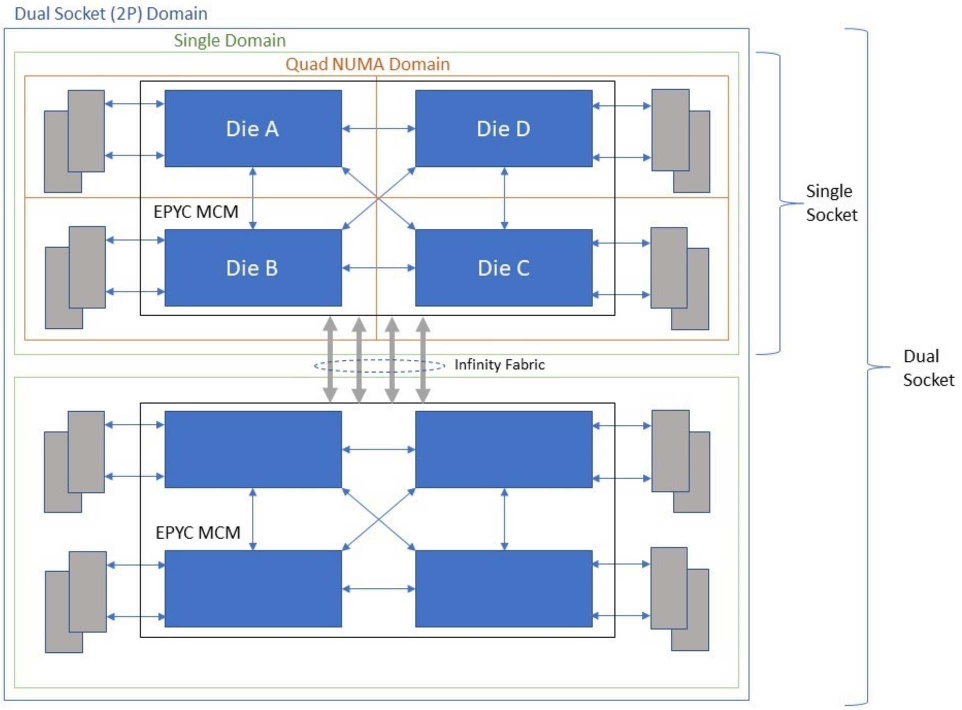
AMD Design Choices

The company built the EPYC processors using multiple Zeppelin die for scalable server systems. A multi-die approach is advantageous, because it allows AMD to build a server processor with a high core count, memory bandwidth, and I/O without a more expensive and low-yielding chip manufacturing process. The resulting MCM solution is also designed to be scalable, allowing AMD to offer a full range of server processors and a leading-edge high performance PC processor (Threadripper), plus a variation of the Zeppelin die that was used as a high-performance PC processor (Ryzen).

The EPYC package consists of four 213 mm^2 Zeppelin die. The aggregate of those four die is 852 mm^2 of silicon area per package. That die size is actually too big to build using today's optical lithography techniques. AMD estimates that if EPYC was built as a (hypothetical) monolithic die, it could remove some of the inter-die IF and PHY, and some additional logic for a ~10% size savings. Removing about 10% from the 852 mm^2 theoretical die reduces it to about 777 mm^2, which can fit inside an optical reticle. Still the 777 mm^2 die would have relatively low yields, because there is an inverse-exponential reduction in yield with larger die size. Using AMD's historical yield model and production defect density, AMD estimated that the four smaller die were less than 60% of the cost of the one large die (Figure 2). Using multiple smaller die has an inherent higher yield and, thereby, a cost advantage.

The tradeoff for using multiple die is that there is additional latency for memory access between the die across the package. The Infinity Fabric connections between packages (Figure 3) are distributed across the four Zeppelin die for balance. Modern operating systems support NUMA and manage the variable latencies, as explained below.

Figure 3: EPYC NUMA Domains



Sources: AMD & TIRIAS Research

A modern server or cloud operating system (OS) will schedule software tasks on a specific core or pool of cores on the same processor (called "processor affinity") and then load data for those tasks in the closest memory to those cores (called "memory affinity"). Hypervisors allocate pools of cores and memory to their guest OS instances but leave the scheduling tasks to their guest OS instances. Containers include their own NUMA-aware schedulers.

Some recent OS research is focused on flipping processor and memory affinity around, by loading data into a specific memory location and then running an associated task in a nearby pool of cores.[1,2] The aforementioned is one example of how the server software community is continuously innovating around NUMA architectures.

NUMA is not limited to AMD processors. Intel servers also use NUMA to optimize performance across multiple sockets and even variable memory latencies within one die. For example, Intel's recently released Skylake-X and Skylake-EP processors have a new on-die mesh network. This mesh network induces variable latencies as cores access memory caches across the die and as they access memory across Intel's QuickPath Interconnect (QPI) socket-to-socket interconnect. Skylake-X and Skylake-EP also have an optimization called sub-NUMA cluster mode (SNC).

SNC mode is useful when threads running on the chip can be grouped and affinitized to specific sections of tiles and they mostly access their own data.[3]

[1] https://www.computer.org/csdl/mags/mi/2016/01/mmi2016010006.pdf
[2] https://www.cs.utah.edu/~rajeev/pubs/ieeemicro14a.pdf
[3] *Intel® 64 and IA-32 Architectures Optimization Reference Manual*, Chapter 8
https://software.intel.com/sites/default/files/managed/9e/bc/64-ia-32-architectures-optimization-manual.pdf

By partitioning one socket into multiple NUMA nodes, sub-NUMA optimization can keep tasks on cores close to the memory controller, just like EPYC. NUMA reduces memory latencies and reduces cross-die data traffic, because Intel's Manhattan Mesh (Figure 4) has no diagonal connections.

In an ideal software runtime environment, every processor socket would have enough memory to satisfy all the threads the OS schedules in each socket. But that is not the case for many modern workloads. Sometimes a data set is too large to be contained in the memory attached to a single socket, no matter that cores allocated to process the data can reside in one socket. Sometimes, workloads compete for local resources and create local bottlenecks.

With dual socket configurations, latency for memory access between sockets will have a significant latency penalty when memory accesses cross a socket-to-socket interconnect, whether that interconnect is AMD Infinity Fabric or Intel QPI. With dual-socket designs, from either AMD or Intel, a NUMA scheduler should place threads and data on cores in the same socket to reduce latencies, otherwise data requests between sockets results in higher latencies.

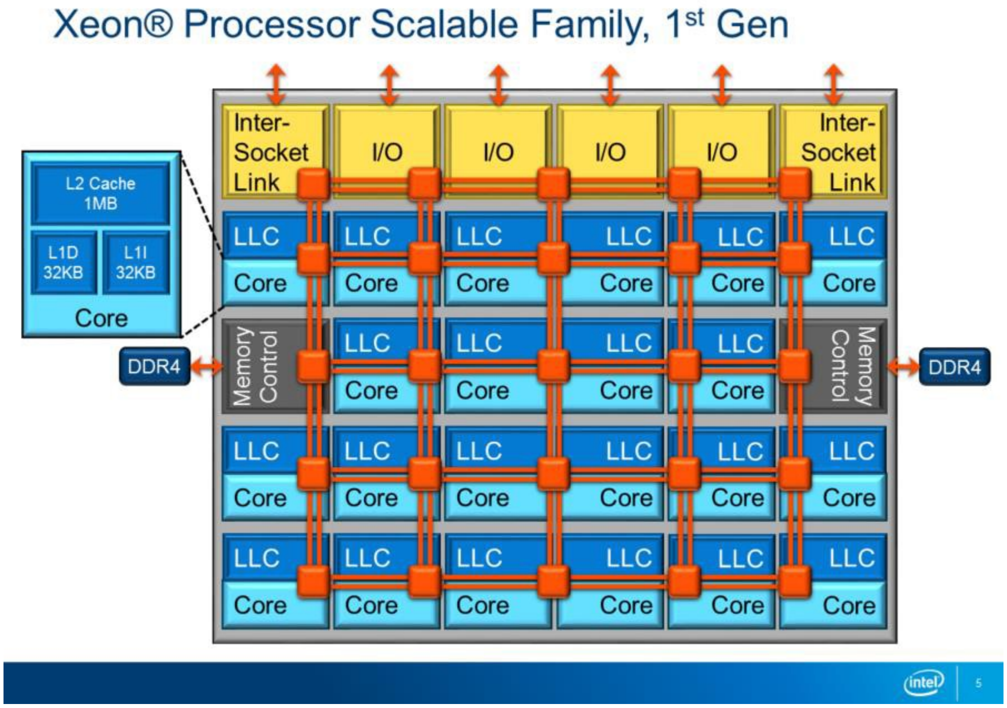
What is Non-Uniform Memory Access (NUMA)?

A multiprocessing (multi-die) architecture in which each processor is attached to its own local memory (called a NUMA domain) but can also access memory attached to another processor.

It is called "non-uniform" because a memory access to the local memory has lower latency (memory in its NUMA domain) than when it needs to access memory attached to another processor's NUMA domain.

The advantage of this architecture is that it provides multiprocessor scalability, adds more memory bandwidth with the addition of more processor, and reduces the memory contention for CPUs if they are competing for access across a common bus (shared front-side bus).

Figure 4: Intel Skylake-X Mesh Fabric



Source: Intel

EPYC: Designed for Servers

AMD designed EPYC and Zeppelin as a server solution first. AMD expects EPYC to excel at VM hosting, virtual desktop interface (VDI), memory-intensive and GPU-accelerated high performance computing (HPC), data analytics, and software defined storage (SDS) workloads. The EPYC processor is expected to excel at scalable workloads with up to eight cores per VM and NUMA support for core / memory affinity but can be used for many different workloads.

AMD took a different design approach than Intel's Broadwell (and previous Xeon processors), which generated some concern over how EPYC would compete with those Intel products.

For NUMA-friendly workloads, AMD EPYC offers similar memory latency but much higher memory bandwidth limits. EPYC's partitioned last-level cache has a significantly lower near-cache latency, and it still is scalable up and down. For workloads that are less NUMA-friendly, EPYC offers better scaling for high-bandwidth workloads but may not be as optimal on some lightweight workloads.

With NUMA-friendly workloads, EPYC 7601 has equivalent loaded latency when compared to Intel's Broadwell (E5-2699A v4) processor (Figure 5). But Broadwell has a memory bandwidth limit of ~65 GB/s (for 1 socket), at which point the latency rises precipitously. AMD's EPYC has much lower latency after the Intel processor hits its single socket bandwidth limit, and EPYC has a higher memory bandwidth limit of ~145 GB/s (for one socket).

For workloads that are less NUMA-friendly, EPYC memory latencies can be up to ~40ns more than Broadwell. But once Broadwell hits its memory bandwidth limit of ~65 GB/s (for 1 socket), the EPYC processor will have the latency advantage until it reaches its memory bandwidth limit of ~145 GB/s (for 1 socket).

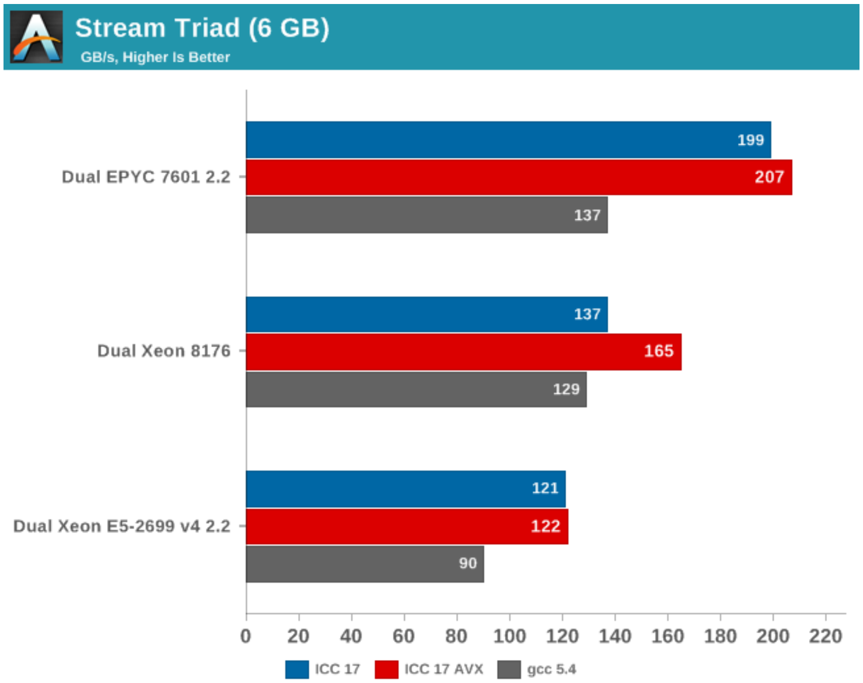
The EPYC architecture is designed to offer many advantages. With up to 32 CPU cores per chip (each dual threaded for up to 64 threads), eight DRAM channels, and 128 PCIe lanes (available no matter the number of cores), EPYC has ample bandwidth and I/O for handling greater virtual machine (VM) density. Those VM instances can also be pinned to NUMA domains for localized access to core, memory, and I/O.

For applications such as in-memory database and analytics, such as Spark and VoltBD, the large memory DIMM capability allows large data sets to be held in local memory. With more PCIe lanes than comparable Intel Xeon processors, EPYC can connect to multiple GPUs and NVMe flash memory drives, without adding an external PCIe switch chip. HPC applications such as CFD, crash simulation, weather modeling, seismic and reservoir modeling should all scale well with the high memory bandwidth. Even Skylake-SP is limited to 768 GB per socket, while EPYC can support 2 TB.

EPYC is designed to do well in many memory bandwidth intensive workloads such as HPC workloads where Intel's Broadwell has a memory bandwidth limit of ~65 GB/s (for 1 socket), at which point it will dramatically increase loaded latency, well beyond EPYC's loaded latencies. The Next Platform tested the Skylake-SP (Xeon Platinum 8180) against Broadwell, which showed a 50% increase in memory bandwidth using the STREAM Triad suite of memory bandwidth tests. The EPYC architecture memory bandwidth limit of ~150GB/s is roughly twice that of Broadwell and is still higher than Skylake-SP.

Testing at Anandtech.com also confirms the advantage of AMD's EPYC over Intel's Skylake-SP (Xeon Platinum 8176) using the Stream 5.10 benchmark. Anandtech compiled the benchmark with two different compilers—the Intel compiler (ICC) and GCC 5.4 compiler. Anandtech also tested the Intel compiler both with AVX instructions enabled and not enabled (Figure 5).

Figure 5: Comparison of EPYC Memory Bandwidth



Source: Anandtech, July, 2017

Notes from Anandtech testing:

> *"The DDR4 DRAM in the EPYC system ran at 2400 GT/s (8 channels), while the Intel system ran its DRAM at 2666 GT/s (6 channels). So the dual socket AMD system should theoretically get 307 GB per second (2.4 GT/s* 8 bytes per channel x 8 channels x 2 sockets). The Intel system has access to 256 GB per second (2.66 GT/s* 8 bytes per channel x 6 channels x 2 sockets)."*

While AMD has often been wary of using ICC for benchmarks, the results showed that ICC could push memory bandwidth harder than GCC and did not disadvantage EPYC at all. Anandtech felt its results, while not as highly tuned as AMD and Intel might produce, were more realistic. With plenty of active threads, EPYC's eight channels of DDR4 2400 memory gave it a 25% to 45% bandwidth advantage over Intel.

While AMD has the advantage of raw bandwidth per socket, Anandtech found the amount of bandwidth to each CCX (four core cluster) is more limited and going off a Zeppelin die increases latency. Using the TinyMemBench, Anandtech measured average time for random memory accesses with buffers of different sizes. The test found AMD's unloaded latency competitive with buffer sizes of 8 MB and smaller, but latencies increase in larger buffers sizes. Techniques such as core pinning and memory interleaving can better optimize workloads for EPYC. The key to extracting the most performance from EPYC may require some system application tuning.

Conclusion

The system design advantages of EPYC appear to be significant for many workloads, including high core and thread count, large memory address space, top memory bandwidth, and industry leading I/O. The use of NUMA is not an impediment to high performance workloads and is a common optimization in server systems today.

The information contained in this report was believed to be reliable when written but is not guaranteed as to its accuracy or completeness.

Product and company names may be trademarks (™) or registered trademarks (®) of their respective holders.

The contents of this report represent the interpretation and analysis of statistics and information that is either generally available to the public or released by responsible agencies or individuals.

AMD, the AMD logo, EPYC, Ryzen, Threadripper, and combinations thereof are trademarks of Advanced Micro Devices, Inc.