The Arm Neoverse N1 Platform: Building Blocks for the Next-Gen Cloud-to-Edge Infrastructure SoC

A. Pellegrini, N. Stephens, M. Bruce, Y. Ishii, J. Pusdesris, A. Raja, C. Abernathy,
J. Koppanalil, T. Ringe, A. Tummala, J. Jalal, M. Werkheiser, A. Kona

White Paper

Abstract— Recent years have seen an explosion of demand for high-performance, high-efficiency compute available at scale. This demand has skyrocketed with the move to the public cloud and 5G networking, where compute nodes must operate within strict latency constraints and power budgets. The Neoverse N1 platform is Arm's latest high-end offering from a scalable portfolio of IP for high performance and energy efficient machines.

NEOVERSE N1 is the new platform of Arm IPs that enables partners to develop systems with competitive performance and world-leading power efficiency across a wide range of markets. On one end of the spectrum, the Neoverse N1 platform is well suited for high performance systems with up to 128 cores organized on an 8x8 mesh. At the same time, customers targeting deployments with strict power and area constraints can rely on Neoverse N1 to create high-efficiency and high-performance systems composed of a dozen or fewer general-purpose cores.

The Neoverse N1 core implements the v8.2-A A32, T32, and A64 Arm instruction sets and includes many infrastructure-focused improvements such as security features, Virtualization Host Extensions, Large System Extensions and RAS Extensions.

Our projections and silicon measurements show that the Neoverse N1 core performs at least 1.6x better for most workloads than Arm's previous design deployed in infrastructure –the Arm Cortex-A72 – with some cloud-native workloads performing up to 2.5x faster. This important speed up was achieved without compromising our best in class power-efficiency. Additionally, scalability significantly improved thanks to a completely redesigned coherent mesh, cache hierarchy, and system IP backplane. As a result, Arm's silicon partners using Neoverse N1 in their designs have numerous opportunities to organize and optimize these components to satisfy their general-purpose compute needs and can take advantage of the many connectivity options for tightly coupling accelerators through technologies such as AMBA, CCIX, and PCIe.

Architecture

The Neoverse N1 cores implement many of the recent extensions to the base Armv8-A architecture that were introduced to improve the performance, introduced to improve the performance, scalability, robustness and security of highly virtualized server and network infrastructure workloads on many-core processors [1].

These architecture extensions include support for dedicated instructions to accelerate inference ML workloads through support of IEEE half-precision floating-point (FP16) and int8 dot product instructions. Neoverse N1 also includes the new CRC32 instructions to accelerate storage applications. Additionally, particular focus has been placed on finer handling of Arm's relaxed memory ordering through the LDAPR instruction (load with ordering semantics similar to Load-Acquire, Store-Release) the introduction of Limited Ordering Regions, support for atomic instructions, and support for persistent memory.

We also included several features to significantly harden Neoverse N1 against known security vulnerabilities:

- Privileged Access-Never (PAN): protects the OS kernel from being "spoofed" into reading or writing user code or data on behalf of malicious programs
- Unprivileged Access Override (UAO): allows the OS kernel to more efficiently manage user code sections that are marked as execute-only for protection
- Stage 2 execute-never: allows the hypervisor to prevent an OS kernel and/or application from executing pages containing writable data, to prevent some exploits
- Side-channel protection: introduces a range of new speculation controls, speculation barrier, and prediction restriction instructions that allow software to mitigate micro-architectural side-channel attacks on speculative execution across different execution contexts

Virtualization is the backbone of much of the modern IT infrastructure, and Neoverse N1 implements enhancements to extend virtualization support and reduce its overhead:

- Hardware update of access/dirty bits: automatically updates status bits in page table entries, avoiding a trap to the OS or hypervisor
- VMID extension to 16 bits: increases the maximum number of simultaneously active virtual machines supported by the address translation system to 65536
- Virtual Host Extension (VHE): more efficient support for Type 2 (hosted) hypervisors, such as KVM, building on the Type 1 (native) hypervisor support already introduced in base Armv8-A

Finally, we extended our performance monitoring infrastructure to support enhanced PMU events and statistical profiling.

Neoverse N1 Core

The Neoverse N1 core is designed to achieve high performance while maintaining the Performance Power Area (PPA) advantage point established with Cortex-A72. To achieve this goal, the team designed the microarchitecture from scratch and focused on features to enhance infrastructure-focused many-core CPU performance.

Neoverse N1 supports an aggressive out-of-order superscalar pipeline and implements a 4-wide front-end with the capability of dispatching/committing up to 8 instructions per cycle. The core deploys three ALUs, a branch execution unit, two Advanced SIMD units, and two load/store execution units. The minimum misprediction penalty is 11-cycle, and we introduced many optimizations in order to preserve a short pipeline without losing power efficiency.

The next sections describe the detail of Neoverse-N1 core microarchitecture. The first two sections describe the core front-end and back-end. The following sections detail the interaction of the core with the memory subsystem, security features, and features added to target the infrastructure market. This section will conclude with a few figures of merit about the core implementation.

Core Front-end

The Neoverse N1 core can fetch up to 4 instructions per cycle to feed its high-performance back-end. One of the biggest improvements from Cortex-A72 is a decoupled branch prediction, which realizes a branch predictor directed prefetch, where the branch predictor can run ahead even if the front-end pipeline is waiting for instruction cache (I-cache) miss refill responses. Even if the I-cache pipeline is stalled on an instruction fetch miss, speculative fetch addresses provided by the branch predictor can continue to access the I-cache and resolve misses through early prefetches.

The branch predictor employs a large 6K-entry main branch target buffer with 3-cycle access latency to retrieve branches' target addresses without accessing the I-cache. Such a sizeable BTB unit helps maintain target history for a large number of branches, which benefits cloud and server workloads with large instruction footprints. The predictor also employs a 64-entry micro-BTB and a 16-entry nano-BTB to minimize bubbles in the front-end. Neoverse N1 also significantly improves both latency and accuracy of the indirect branch prediction algorithm. The branch direction predictor is also optimized to target behaviors observed on many server workloads: once a prediction is made, the predicted address is stored into a 12-entry fetch queue which tracks future fetch transactions.

Once the branch predictor creates a next fetch address, the address is fed into a fully associative 48-entry instruction TLB and a 4-wayset-associative 64KB I-cache to read out the instruction opcode. The I-cache can deliver up to16B of instructions per cycle.

Since the branch predictor has higher bandwidth than the I-cache, unless the pipeline was recently flushed due to a branch misprediction, the fetch queue typically holds a few pending transactions. To mitigate branch misprediction penalty, I-cache reads areoverlapped with I-cache tag matching. After the fetch queue reaches a threshold in the number of fetch transactions, the I-cache read operation is serialized to maximize efficiency. The NeoverseN1 core can support up to 8 outstanding I-cache refill requests to the higher cache hierarchy.

The stream of fetched instructions is then forwarded to a 4-wide decoder, where an instruction may be cracked into multiple simpler internal macro-operations. Each decode lane can decode one Arm instruction per cycle, and the most frequently used instructions (e.g., simple ALU, branch and load/store) are decoded as a single macro-operation. To simplify and speed-up the decode process, the I-cache can store partially decoded instructions.

Core Back-End

Decoded instructions are renamed before being dispatched to the out-of-order engine. The renaming unit can receive up to 4 macro-ops per cycle. Each macro-op can be cracked to up to 2 micro-operations during the renaming process. Therefore,up to 8 micro-operations can be dispatched into the out-of-order engine each cycle. Additionally, the rename unit can automatically eliminate simple register-to-register data movement instructions through its rename tables.

Once the micro-ops are dispatched, instruction status is tracked in the commit and the issue queues. The commit queue can track up to 128micro operations. The commit unit tracks a dispatched instruction until all prior instructions are committed, and up to 8 micro-ops can be committed per cycle.

The issue queue tracks the availability of source operands required to execute corresponding micro operations. When all its source operands are available, an instruction is picked and issued to the correct execution pipeline. Neoverse N1 supports a distributed issue queue with more than 100 micro- operations to increase the overall out-of-order window size. When the issue queue is empty, dispatched instruction can bypass such a queue to minimize latency.

Neoverse N1 employs multiple pipelines for each type of instruction: 4 integer execution pipelines, 2 load/store pipelines, and 2 Advanced SIMD pipelines. As needed, each pipeline can forward its results to the others.

Memory Subsystem

The memory architecture for Neoverse N1 is designed to enable larger, faster and more scalable caches than its predecessors. The 64kB 4-way set associative L1 data cache

(D-cache) has a 4-cycle load to use latency and a bandwidth of 32 bytes/cycle. The core-private 8-way set associative L2 cache is up to 1MB in size and has a load-to-use latency of 11 cycles. The Neoverse N1 core can also be configured with smaller L2 cache sizes of 256kB and 512kB with a load-to-use latency of 9 cycles. The L2 cache connects to the system via an AMBA 5 CHI interface with 16-byte data channels. The Neoverse N1 core can directly interface to the mesh interconnect enabling minimum latency to the system-level-cache and DRAM. Alternatively, multiple cores can be configured in a cluster of cores containing a snoop filter and an optional L3 cluster cache. The cluster cache can be up to 2MB, with a load-to-use latency ranging between 28 and 33 cycles, depending on the configuration. A Neoverse N1 SoC can support up to 256MB of shared system-level cache. In the event none of these caches are effective at filtering a requested memory address, the Neoverse N1 core employs a "cache-miss" predictor which bypasses the whole cache hierarchy and all snoop filters to issue a "Prefetch Target" request to compatible memory controllers, reducing the incurred miss latency.

Neoverse N1 employs a next generation data prefetcher, which is similar to the one deployed in the Cortex-A76 core, but with key improvements for large scale systems. This updated prefetcher achieves high coverage and accuracy on a variety of access patterns ranging from simple streams and strides to sophisticated spatial patterns. Such a prefetcher coordinates requests to multiple levels of cache and across virtual memory pages, preloading both TLBs and caches. Finally, multiple cache replacement policies were designed and tuned to work in coordination with these prefetchers, resulting in our first prefetch-aware replacement policy.

Security Features

During the development of the Neoverse N1 core, side channel attacks exploiting speculative execution [2,3] were reported and several architectural and microarchitectural mitigations were introduced to address these security vulnerabilities.

Neoverse N1 implements some of the Arm v8.5 architecture features such as the SSBS (Speculative Store Bypass Safe) control bit, and the SSBB and PSSBB (Speculative Store Bypass Barrier) instructions. These newly introduced barriers allow software to actively protect against Spectre Variant 4 exploits by preventing load instructions from returning data written to a matching virtual or physical memory location by speculatively executed store instructions prior to the barrier [4].

Spectre Variant 2 attempts to exploit the branch predictor by injecting branch targets that cause the victim process to speculate through a specific code path. To address this threat, we designed a hardware mechanism to prevent consumption of malicious target injections [4]. Malicious software cannot inject branch target information to control speculative behavior of the victim process since the branch predictor in the Neoverse N1 core prevents a process from using the predicted branch trained by a different process.

Infrastructure Focused Core Features

Server systems targeting modern cloud deployments typically require multi-socket platforms with high core counts and large memory capacity: Neoverse N1 implements a number of features targeted to this class of machines.

One of these features is the support for hardware coherent I-caches. Like other architectures, the Arm architecture does not require I-caches to maintain coherent through hardware mechanisms. Hence, on legacy Arm systems, software must issue the necessary cache maintenance operations whenever memory containing instructions is modified. Typically, these invalidations are broadcast to all cores within the same coherency domain to realize transparent instruction memory access. Unfortunately, such broadcasts can limit scalability on high core count systems. Neoverse N1 eliminates this bottleneck by implementing a fully hardware coherent I-cache that requires no software maintenance and leverages the same hardware coherency mechanisms utilized by the data cache. To ensure software compatibility, unnecessary I-cache maintenance operations still issued by legacy software are treated as 'no operation' in the core. For more recent software, Neoverse N1 includes a status bit that allows users to discover that the I-cache is hardware coherent, hence cache maintenance instructions are not necessary.

Another infrastructure feature added by Neoverse N1 is the support for the Arm v8.2 RAS architecture. The RAS architecture provides a framework for detecting, classifying, and reporting errors that is consistent across all components of the SoC. Neoverse N1 also adds the ability to defer errors from one component to another. For example, an uncorrectable data ECC error in DRAM can be propagated to a core which can cache the corrupted data but flag the erroneous word as 'poisoned'. When an instruction such as a load consumes the poisoned data, it generates an exception. If the poisoned data is never consumed by the core, the data will eventually be evicted from the core caches but will retain the poison information in the system-level-cache or DRAM.

Finally, Neoverse N1 increases the ASID and VMID widths to 16 bits, allowing for more guest operating systems and applications within each guest. Neoverse N1 also extends the physical address width to 48-bit, allowing systems to support up to 256TB of physical memory.

Implementation

The Neoverse N1 core design was fully evaluated within our internal development environment. Figure 1 shows the core floorplan of our reference implementation, which employs a 64KB I-cache, a 64KB D-cache, and 1MB L2 cache.

Our optimized physical implementations indicate that the core and L2 cache can reach an operating frequency of 3.1GHz. When executing an intense integer workload, the estimated power consumption for a 7nm implementation of a core is 1.0W and 1.8W when clocked at 2.6GHz and 3.1GHz, respectively. The core area is estimated to be 1.15mm^2 for a 512kB L2 configuration and 1.40mm^2 for a 1MB configuration. Our models project that a 64-core reference system can achieve 190 SPECint2017 rate (estimated). For such a system, the total SOC power is projected to be 105W.

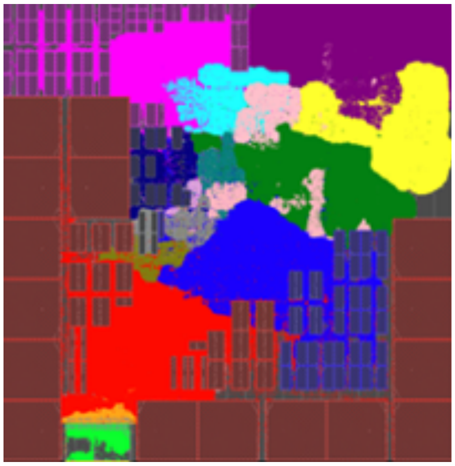

Figure 1:
Neoverse N1 core floorplan used for our reference design, which includes 64KB I-cache, 64KB D-cache, and a 1MB core-private L2.

Neoverse N1 Coherent Mesh Interconnect (CMN-600)

The CMN-600 product family is Arm's second- generation, highly configurable, mesh-based coherent interconnect based on CHI cache coherent protocol specification. CHI is a packet-based, point- to-point, topology agnostic, layered architecture protocol. The coherent interconnect is a vital component to enable many-core systems to scale without compromising latency and available memory bandwidth. A mesh topology was chosen to address those challenges and is designed to address those challenges and is designed to support one clock cycle delay per hop. CMN-600 can scale from a 1x2 mesh to a 8x8 mesh and is designed to operate at up to 2GHz clock frequency. Customers can configure mesh size, topology, and bisection BW to match the architecture that best fits their PPA targets.

CMN-600 includes a distributed set of fully coherent home nodes which are software-configurable hash address-interleaved. Software-configurability allows customers to support different hash interleaving granularity, with the minimum interleave being 64Bytes. Such configurability enables traffic distribution and traffic isolation based on the characteristics of the targeted applications and allows affinity-based system cache groups (SCG) allocation, which helps with traffic localization in bigger systems.

Each HN-F slice includes a snoop filter and a system level cache (SLC) with enhanced replacement policies. System architects that adopt the Neoverse N1 platform can choose the number of HN-F slices to deploy based on system cache capacity needed and system bandwidth requirement, and total SLC capacity can range from 0MB to 256MB.

The SLC is a victim cache for core clusters with adaptive cache allocation based on data-sharing detection. SLC also acts as a DRAM cache for IO Requestors (PCIe, DMA etc.) with a smart allocation policy. In addition, the SLC supports software programmable source-based cache capacity control which mitigates the "noisy-neighbor" shared system cache thrashing problem.

Some key performance enhancement features of CMN-600 include:

- Support for Arm and PCIe architecture atomic transactions at the home nodes. This allows atomic transactions to be issued by the cores to the home nodes, where they are resolved. The capability to execute far atomics operations improves performance for contended variable updates

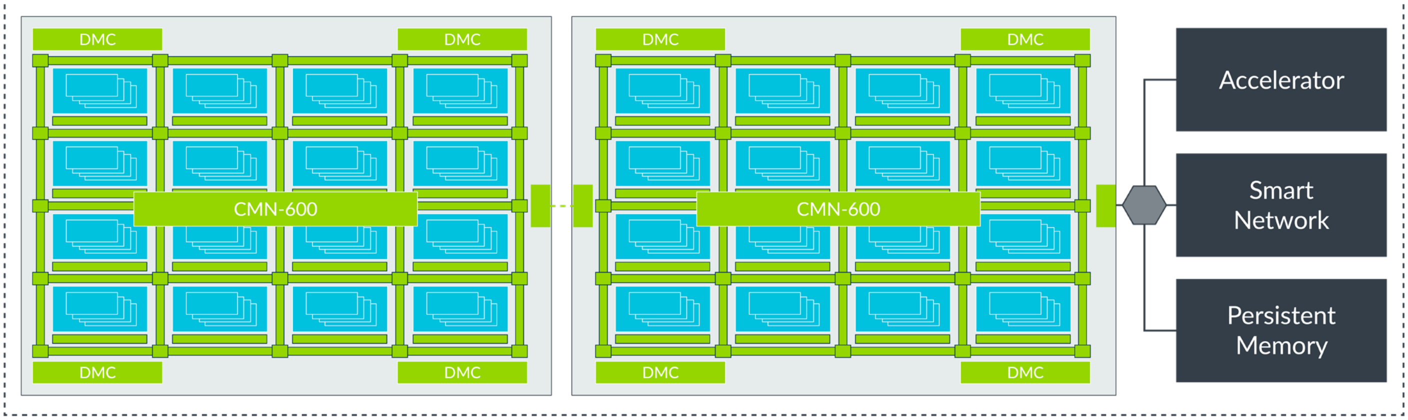

Figure 2: CMN-600 based scale-up server node with two compute dies and acceleration

- Prefetch hint can be issues from a Neoverse N1 core directly to the Memory Controllers in order to minimize DRAM latency on cache misses. Other features to reduce data latency include: Direct Memory Transfer from Memory Controllers to requesting cores and Direct Cache Transfer from peer cores to the requesting core. In aggregate, we estimate these features to reduce data latency on the interconnect by up to 37%
- Cache Stashing, which allows an IO peripheral such as a PCIe endpoint to place incoming data on various levels of the cache hierarchy (SLC, L3, L2) to enable quicker access to this data. When SLC stashing is enabled, silicon measurements show improvements up to 33% packet/second on a single core and up to 60% on multicore tests for DPDK L3 Forwarding tests. Further improvements are expected for applications that can stash IO data in the cores' private L2 or in the core cluster L3

CMN-600 is designed to support high throughput IO traffic from various requesters such as DMA, PCIe, etc., and can achieve full PCIe Gen4 upstream and downstream bandwidth. PCIe or DMA writes can be stashed in the SLC or directly into the core caches. Direct stashing to CPU caches allows improved performance and avoids SLC pollution.

CMN-600 provides at-speed self-hosted debug and trace capabilities with distributed debug monitors within the interconnect. Our interconnect supports programmable transaction tagging and tracing, which can be used for statistical profiling and end to end latency breakdown analysis.

Multi-chip support using CCIX

CMN-600 supports CCIX protocol (Cache Coherent Interconnect for Accelerators) to coherently connect hardware accelerators such as GPUs, smart NICs, smart storage, FPGAs, DSPs etc. to CMN-600 based host node. By extending the benefits of full cache coherency to these hardware accelerators, Neoverse N1 enables true peer processing with shared memory, which also eliminates the need for software to intialize transfers of data

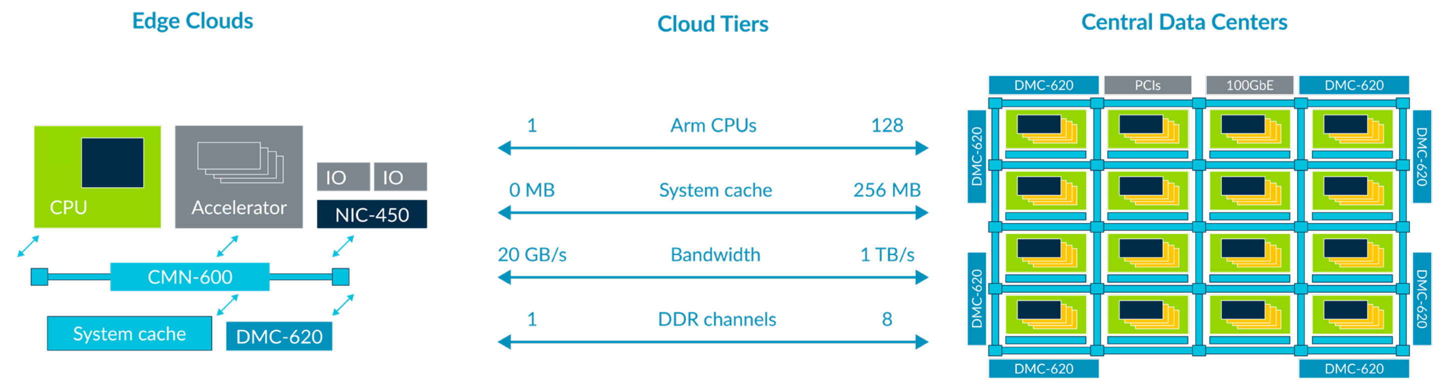

Figure 3:
System Scalability of CMN-600

between devices. CMN-600 also supports CCIX independent memory expansion where the CCIX link is used to communicate with memory on a remote chip.

CMN-600 leverages the same CCIX connection to enable Symmetrical Multi-Processing (SMP) across multiple chips to enable homogenous computing. In order to enable this link for SMP use cases, the CCIX link can be augmented with special features to communicate Arm ISA specific information that is not required for the host-accelerator use case.

Figure 2 shows a scalable system where multiple CMN-600's form a host node for homogenous computing while connecting to hardware accelerators for heterogenous compute use cases.

IO Memory Management and Interrupt Handling

Arm's latest System MMU, MMU-600, supports stage-1, stage-2, and nested address translations with address space mapping and security mechanisms to prevent un-authorized accesses. In a typical system, these two translation stages are managed by the guest operating system and the hypervisor, respectively.

MMU-600 support's PCIe Address Translation Service to allow PCIe-based IO devices or accelerators (masters) to pre-fetch translations well in advance and place them in device-managed Address Translation Caches, hence avoiding the translation overhead in the MMU. Support for the PCIe Page Request Interface further enhances system performance by enabling devices to use un-pinned pages and virtual memory.

The Neoverse N1 platform supports PCIe Root Complexes with Single Root IO Virtualization function, which allows virtualized PCIe functions to be integrated into a system to provide IO virtualization. In a PCIe Root Complex, each virtual function (VF) and physical function (PF) pair mapping is assigned a unique PCI Express Requester ID that is mapped to a unique StreamID in the system to match the Arm architecture requirements. MMU-600 maps virtual addresses to physical addresses using the StreamID pairs. With support for up to 2^{24} StreamID's, MMU-600 allows simultaneous mapping of millions of PCIe VFs. In a virtualized environment, the VF is assigned to a virtual processing element (VPE) and the system traffic flows directly between the VF and VPE. As a result, the IO overhead in the software emulation layer is diminished, significantly reducing the overhead of a virtualized environment compared to a non-virtualized one.

With total bandwidth support of up to 64GB/s per IO interface, MMU-600 is architected to support throughput requirements for next generation PCIe Gen5.

GIC-600 is a GICv3 architectural specification compliant interrupt controller with enhanced support for large number of cores and multiple chip configurations. GIC-600 structurally consists of Interrupt Translation Service (ITS) blocks, a distributor and re-distributors. The ITS block translates PCIe Message Signaled Interrupts (MSI/MSI-X) interrupts to Arm Locality-specific Peripheral Interrupts (LPI). The distributor manages interrupt routing and directs interrupts to the appropriate core-cluster that services the interrupts. In a virtualized environment core interrupts are virtualized, and the incoming physical interrupts are mapped by a hypervisor to a VM. The inter-socket or inter-chiplet messages are ported to the native communication transport protocol supported by the system.

N1 Software Development Platform

In order to create a proof point for our technology, we taped-out a test chip based on Neoverse N1 IPs called the N1 Software Development Platform (N1 SDP). This system consists of four Neoverse N1 cores, configured as two pairs of two-core clusters. Each core has 64KB private L1 I/D cache and 1MB private L2cache. Each cluster connects its two cores through a DynamIQ Shared Unit (DSU), which is configured with a 1MB shared L3 cluster cache. The system is configured with 2 DDR4 3200 memory controllers and 2 PCIe Gen4 Root Complexes, one of which supports CCIX for attachment of cache-coherent IO devices or to support multichip configurations. A 4x2 CMN-600 coherent mesh network connects all the high performance on-chip components.

N1 SDP provides early N1 silicon samples and is a vehicle for software development and evaluation environment to customers and partners.

Real World Performance

We evaluated the performance of N1 systems extensively both pre-silicon as well as in silicon implementations such as N1 SDP. Our projections and silicon measurements show that N1 systems match or exceed performance on currently available cloud instances in many relevant workloads. Single core performance was improved from Cortex-A72 by

65% and 100% on average for integer and floating-point workloads, respectively. System-level performance uplifts are much higher thanks to the multipliers offered by the unprecedented scalability of our CMN-600 mesh interconnect.

Beyond targeting general performance improvements, we spent significant effort optimizing the system for common behaviors observed in server and networking workloads.

For example, a class of workloads we focused on is high-throughput HTTP server such as NGINX. NGINX is a highly concurrent, high performance application that can be used as web server, reverse proxy, and API gateway. Neoverse N1 performance uplifts for this class of workloads is directly related to:

1. Memory latency and bandwidth: up to 2x increase in memcpy bandwidth vs Cortex-A72
2. Context switch: up to 2.5x faster than Cortex-A72
3. Core front-end: significant reduction in branch mispredicts (7x) and cache misses (2x) vs Cortex-A72

These stressors are very common with throughput applications such as MemcacheD and HHVM. Overall, Neoverse N1 can reach 2.5x higher throughput on NGINX static web-server vs. a similarly configured Cortex-A72 based system.

Another class of applications we focused our attention on are runtime frameworks such as Java Virtual Machines and .Net Frameworks. These runtime environments are the foundation of much of the applications running in the cloud and are a natural target for our design. At a high level, on Neoverse N1 we focused on a few relevant stressors for these workloads:

1. Object management: up to 2.4x more memory allocations and 1.6x faster in copying characters on Java microbenchmarks vs Cortex-A72
2. Managing the instruction footprint: on a Java-Based-Benchmark, I-cache miss rate and branch mispredictions was reduced by 1.4x and 2.25x vs Cortex-A72, respectively
3. Process synchronization for garbage collection: locking throughput and latency improved by 2x thanks to the Large System Extensions Arm atomic instructions

We expect to see higher performance gains as Neoverse N1 systems become more broadly available for software optimizations and application tuning [5]. At the time of writing, Arm partners report that initial evaluations of real-world workloads on systems deploying Neoverse N1 show up to 40% better performance compared to similarly configured systems currently on the market.

Conclusions

The Neoverse N1 platform provides Arm's partners with the high-performance IPs necessary to architect a general compute solution for addressing the infrastructure market. These building blocks offer the versatility, performance, features and power-area efficiency to succeed in the infrastructure market. We anticipate high-core count designs based on Neoverse N1 to be deployed in public cloud as an alternative architecture for main compute nodes, enabling lower total cost of ownership for data center operators and edge installations of cloud compute while delivering greater design diversity. We fully expect Neoverse N1 to also find a home in more advanced network, storage, and security appliances as well as on edge compute installations deployed by network operators with design points starting at 8 cores.

Acknowledgments

We thank Mike Filippo, Ann Chin, and the many Arm engineers in Austin, TX and worldwide who contributed to Neoverse N1 definition, architecture, design, physical implementation and software optimizations.

References

1. Arm® Architecture Reference Manual Armv8, for Armv8-A architecture profile Documentation [Online]. Available: https://developer.arm.com/docs/ddi0487/latest

2. Moritz Lipp, Michael Schwarz, Daniel Gruss, Thomas Prescher, Werner Haas, Anders Fogh, Jann Horn, Stefan Mangard, Paul Kocher, Daniel Genkin, Yuval Yarom, and Mike Hamburg "Meltdown: Reading Kernel Memory from User Space", 27th USENIX Security Symposium 2018

3. Paul Kocher, Jann Horn, Anders Fogh, Daniel Genkin, Daniel Gruss, Werner Haas, Mike Hamburg, Moritz Lipp, Stefan Mangard, Thomas Prescher, Michael Schwarz, and Yuval Yarom "Spectre Attacks: Exploiting Speculative Execution" 40th IEEE Symposium on Security and Privacy 2019

4. Arm® Vulnerability of Speculative Processors to Cache Timing Side-Channel Mechanism [Online]. Available: https://developer.arm.com/support/arm-security-updates/speculative-processor-vulnerability

5. Arm® Neoverse™ N1 Software Optimization Guide Documentation [Online]. Available: https://developer.arm.com/docs/swog309707/a

6. The [Arm Neoverse N1 Platform: Building Blocks for the Next-Gen Cloud-to-Edge Infrastructure | https://ieeexplore.ieee.org/abstract/document/8986666/authors] whitepaper has been published by the IEEE Computer Society.

All brand names or product names are the property of their respective holders. Neither the whole nor any part of the information contained in, or the product described in, this document may be adapted or reproduced in any material form except with the prior written permission of the copyright holder. The product described in this document is subject to continuous developments and improvements. All particulars of the product and its use contained in this document are given in good faith. All warranties implied or expressed, including but not limited to implied warranties of satisfactory quality or fitness for purpose are excluded. This document is intended only to provide information to the reader about the product. To the extent permitted by local laws Arm shall not be liable for any loss or damage arising from the use of any information in this document or any error or omission in such information.