Solution Brief

Machine Learning
4th Gen Intel® Xeon® Scalable processors

Optimizing Machine Learning (ML) Models with Intel® Advanced Matrix Extensions (Intel® AMX)

Bidirectional Encoder Representations from Transformers (BERT) model throughput shows 2x–3x performance gains with 4th Gen Intel® Xeon® Scalable processors and Intel AMX versus the previous generation[1,2]

In this solution brief, standard BERT models of 12 layers, 768 hidden size, 12 heads, and 128 sequence length (token size) are used as the proxy model for introduction of the fusion optimization methodology.

Overview

Bidirectional Encoder Representations from Transformers (BERT) is a widely used ML model and technique for natural language processing (NLP). BERT has been used to refresh countless records in NLP tasks since its inception. It has also performed extremely well in practical core-bound applications.

For search, machine translation, man-machine interaction, and other NLP tasks, BERT has been widely adopted across multiple user scenarios. Because BERT performance directly affects the user experience with applications and increases the queries per second (QPS) throughput rate, engineers have considered a wide variety of ways to optimize the model to improve its performance.

Tencent StarLake Lab personnel explore advanced cloud computing, artificial intelligence (AI), security, storage, and network technologies to deliver solutions that improve data center performance and reduce the total cost of ownership (TCO) of data centers. The Tencent Machine Learning Platform Department (MLPD) is the heart of the Tencent AI platform, constantly working to drive innovations across Tencent's internet and technology businesses. The MLPD engages in R&D covering a broad range of fields, including computer vision, voice recognition, graph computation, and NLP. Solutions created by the MLPD have been broadly applied to major scenarios in social media, personalized advertising, gaming AI, and content recommendation and search. BERT plays a key role in applications across all these tech sectors.

Intel has closely collaborated with Tencent MLPD and Tencent StarLake laboratory on BERT inference optimization using Intel® AMX, a built-in accelerator for 4th Gen Intel® Xeon® Scalable processors. The teams demonstrated that BERT model throughput [INT8] could increase 2x and BERT model throughput [BF16] could increase 3x when running on systems powered by 4th Gen Intel Xeon Scalable processors using Intel AMX.[1,2] By combining Intel AMX and software optimizations into a powerful unified solution, Tencent aims to evolve its capabilities to deliver a consistent service experience and to optimize TCO.

Tencent Social Applications Optimization

The Tencent social applications connect over a billion active users around the world. One of most popular Tencent social applications was released in 2011 and became the world's largest standalone mobile app in 2018. In fact, it was nicknamed "China's app for everything" because of its impressive array of functions and uses, which include text messaging, voice messaging, broadcast messaging (one-to-many), video games, and video conferencing. Additionally, it includes photo-sharing, video-sharing, and location-sharing features.

Most users of that app tend to use its integrated search engine to search for text messages, articles, mini-programs, short videos, music, and other popular types of content. The key challenges for the search engine are how to handle large-scale queries and respond promptly with the search results. Deep optimization with Intel AMX is deemed to be the solution to these challenges, to improve the overall application search experience by decreasing TCO and leveraging the existing general-purpose infrastructure of the search engine.

Fusion Optimization

Fusion optimization had previously been realized through the FP32 solution by fusing 12 layers of the BERT base model into a single, large operation (op). Intel AMX now provides the functionality necessary for even more in-depth fusion optimization.

Because MatMul and BatchMatMul ops can consume significant amounts of BERT time, optimizing MatMul and BatchMatMul is key to improving performance. Based on past experience, performance optimization can be achieved by either reducing computation or reducing memory access. Removing unnecessary ops from the model can reduce computation, which in turn will decrease the number of instructions. Merging several ops into one can reduce memory access, which allows for accessed data to be kept in cache until needed for further use.

Based on these ideas, Tencent MLPD and Intel transformed some of the most time-consuming processes into a large operation called "Fused BERT op," as shown in Figure 1.

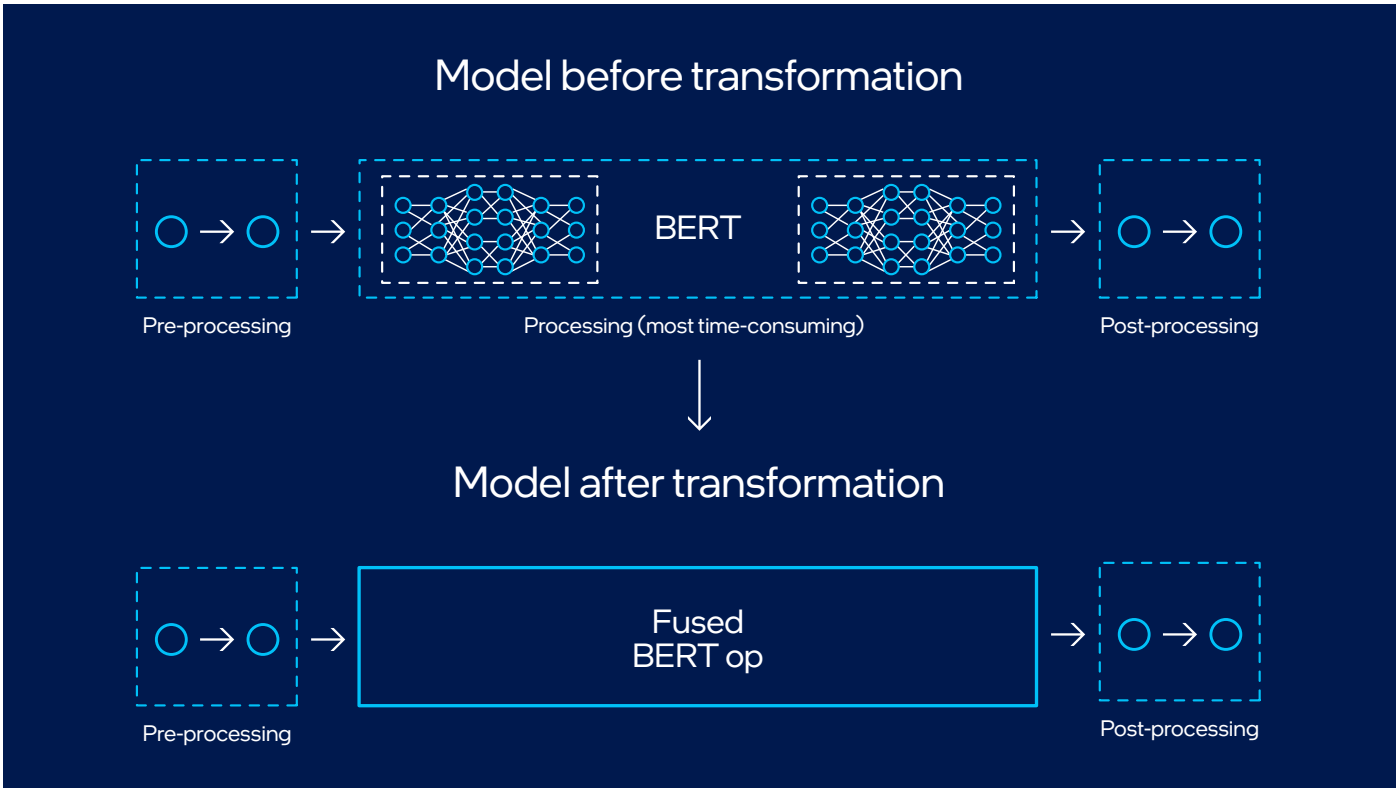

Figure 1. BERT optimization by merging several ops into a single Fused BERT op

The Fused BERT op achieves optimization in the following ways:

- As inputs of query, key, and value (QKV) MatMul are equal, the weights of these ops can be merged into a big weights matrix, and then merged into a big QKV MatMul. After these weights are merged, the memory of each weight and each output is no longer continuous. As such, when the QKV output is used as the input of the next op, a suitable "stride" must be configured. This optimization flow is illustrated in Figure 2.

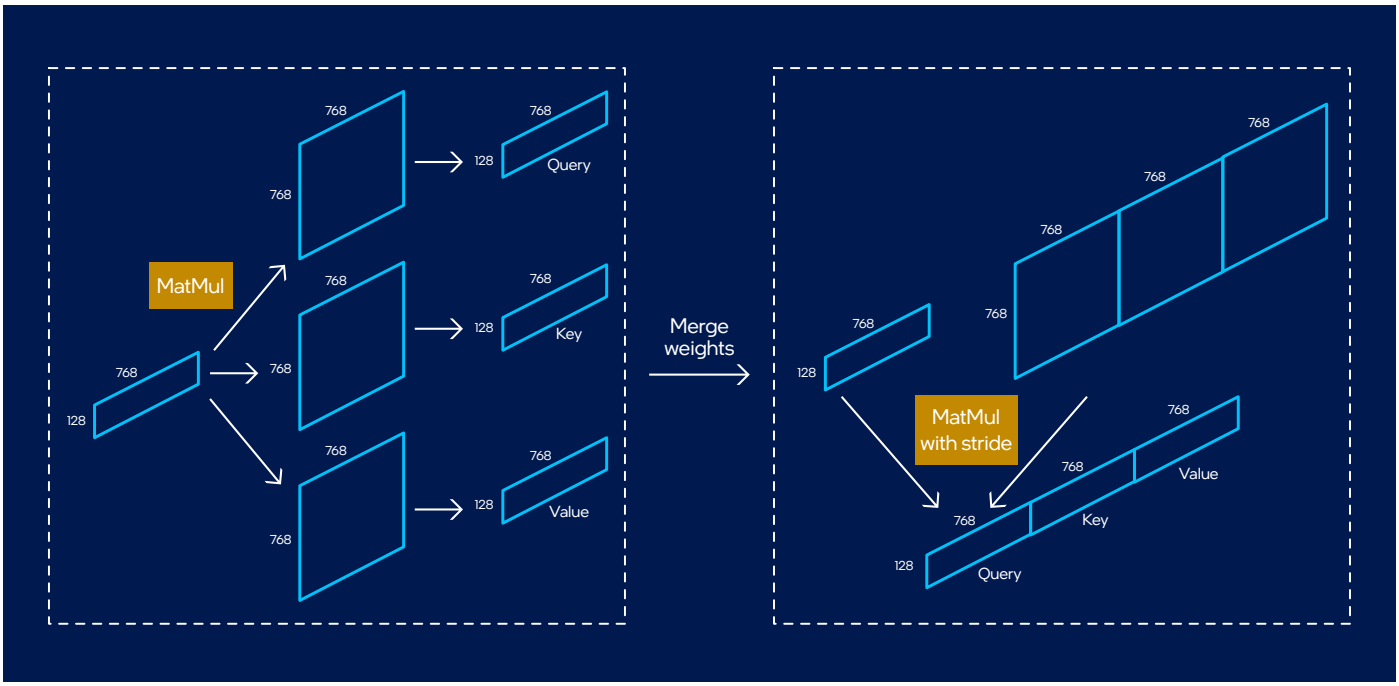

Figure 2. QKV MatMul optimization flowchart

- The transpose ops are removed before and after the BatchMatMul as oneDNN supports BatchMatMul with stride. This saves a large amount of memory access and computation. The optimization flow is shown in Figure 3.

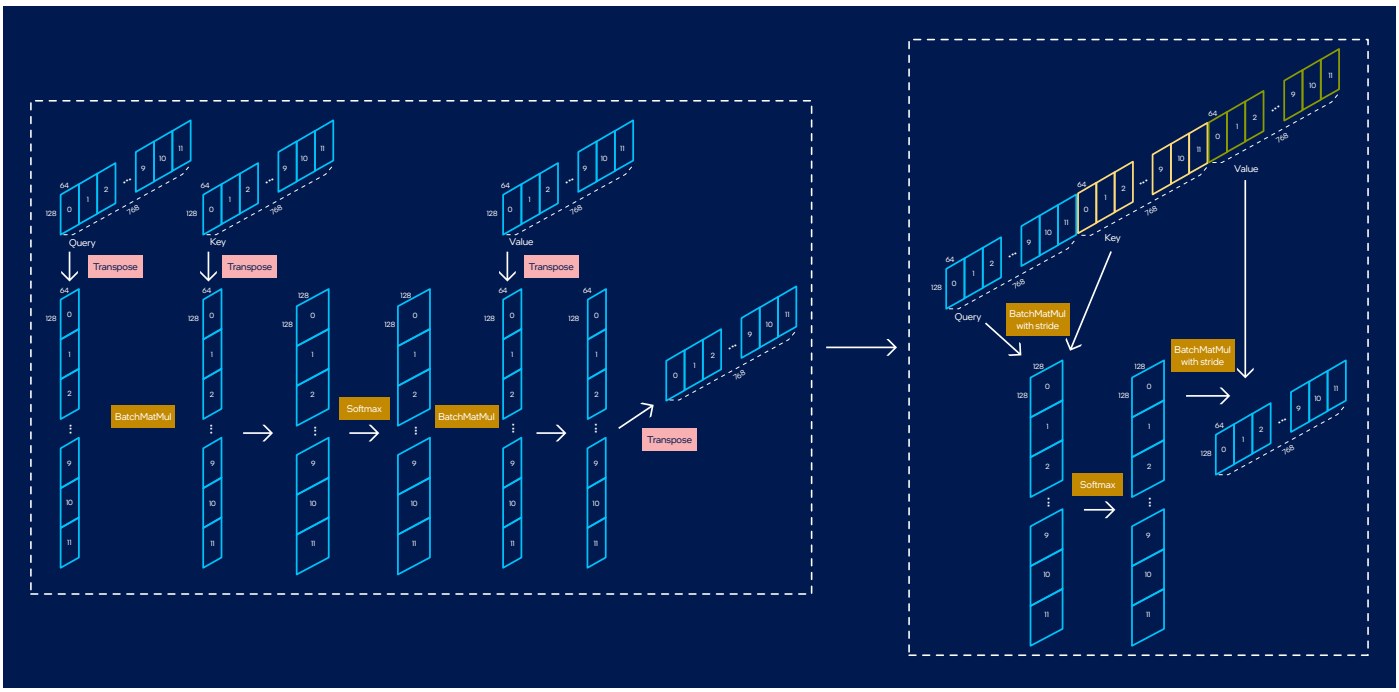

Figure 3. BatchMul optimization flowchart

- Certain operations can be integrated into MatMul primitives, as oneDNN supports MatMul with BiasAdd and certain post ops (such as OutputScale, Sum, Relu, Gelu, and Tanh). Integrating these ops improves cache use efficiency. In other words, this can keep the data warm in the cache so that related data can be called by multiple tasks.

Feature Dense Optimization

In NLP tasks, features or data usually have unequal lengths, and a large amount of padding needs to be inserted to form batches. This results in a large amount of unnecessary computational overhead. Intel partnered with Tencent MLPD in developing the Feature Dense optimization solution for the BERT model, which removes computational overhead and significantly improves task performance. The larger the batch size, the greater the performance improvements. Figure 4, which shows only part of the BERT model, outlines how this optimization works.

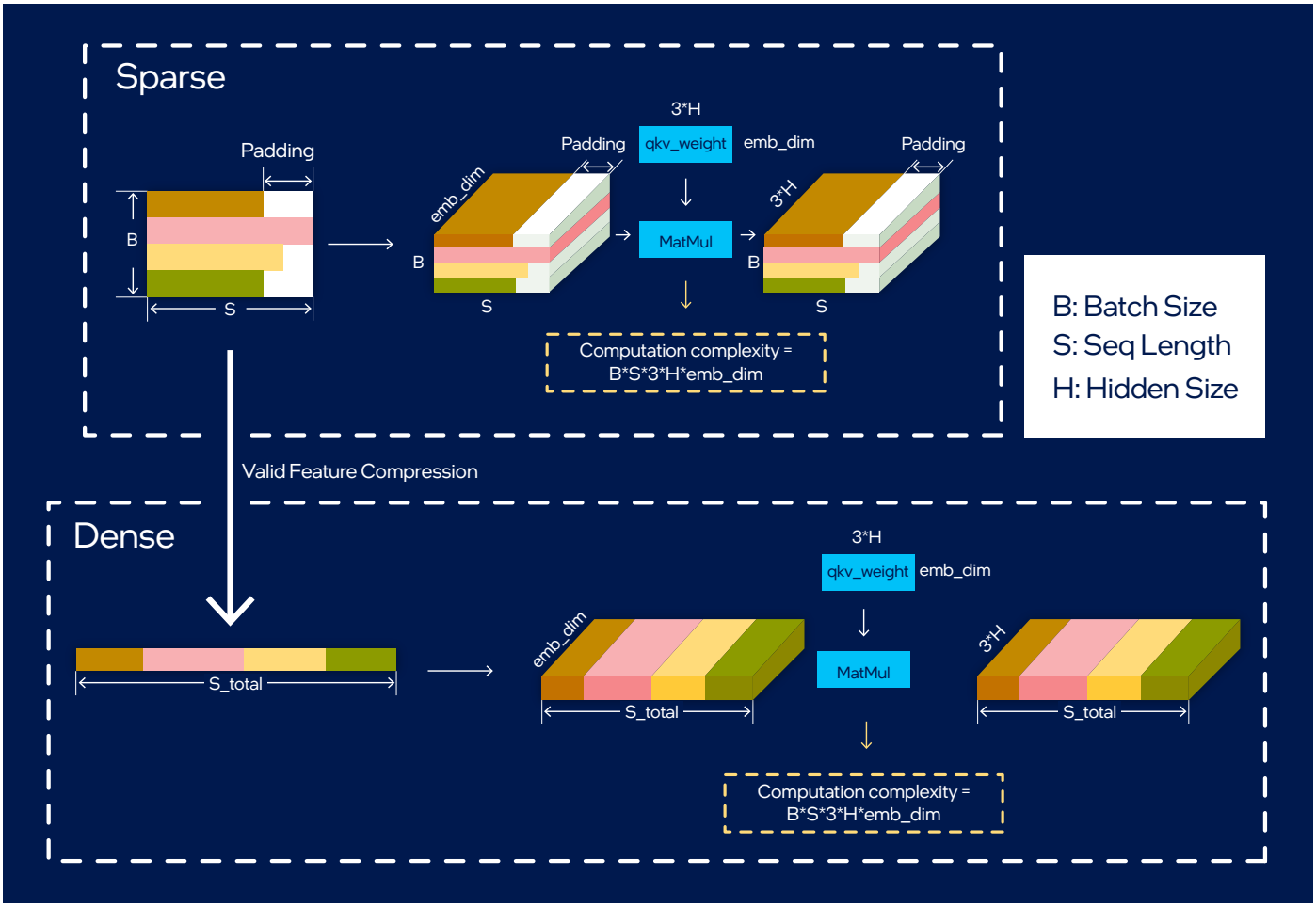

Figure 4. Feature Dense optimization solution

As Figure 4 illustrates, the Feature Dense optimization solution removes padding from the input and connects the data of each batch, one by one, to form a piece of one-dimensional data. Subsequent operations such as embedding, QKV MatMul, and later MatMul ops (omitted from the figure) are based on the compressed data. This greatly reduces the computational complexity, as shown in the figure.

BF16/INT8 Optimization

Application of the FP32 optimization outlined earlier in this solution brief has already improved BERT performance significantly, but there is room for more improvement. Further reducing memory access will result in even better performance. This can be done by reducing the size of data during computation. FP32 data, including input and weights, can be converted to BF16/INT8 data before it is used for computing.

The question then becomes this: Which platform can support BF16/INT8 computation while retaining strong performance? 3rd Gen Intel Xeon Scalable processors (Cooper Lake and Ice Lake) come equipped with Intel® Deep Learning Boost (Intel® DL Boost), which support VNNI (INT8) instructions for vector multiply. Cooper Lake also supports the BF16 numerical format. This enables the leveraging of the FP32 optimization solution for BF16 or INT8 optimization. Test results confirm that BF16 or INT8 optimization can improve performance markedly, compared to the FP32 solution.

Intel AMX on 4th Gen Intel Xeon Scalable Processors

Is there any space for further optimization? Absolutely. Intel AMX is a built-in accelerator of 4th Gen Intel Xeon Scalable processors. Intel AMX provides a 64-bit programming paradigm with a set of two-dimensional registers (tiles) representing sub-arrays from a larger two-dimensional memory image, plus an accelerator capable of tile ops. The first implementation is TMUL, which stands for "tile matrix multiply unit."

Figure 5 shows a conceptual diagram of the Intel AMX architecture. An Intel architecture host drives the algorithm, memory blocks, loop indices, and pointer arithmetic. Tile loads and stores and accelerator commands are sent to multi-cycle execution units—TMUL.

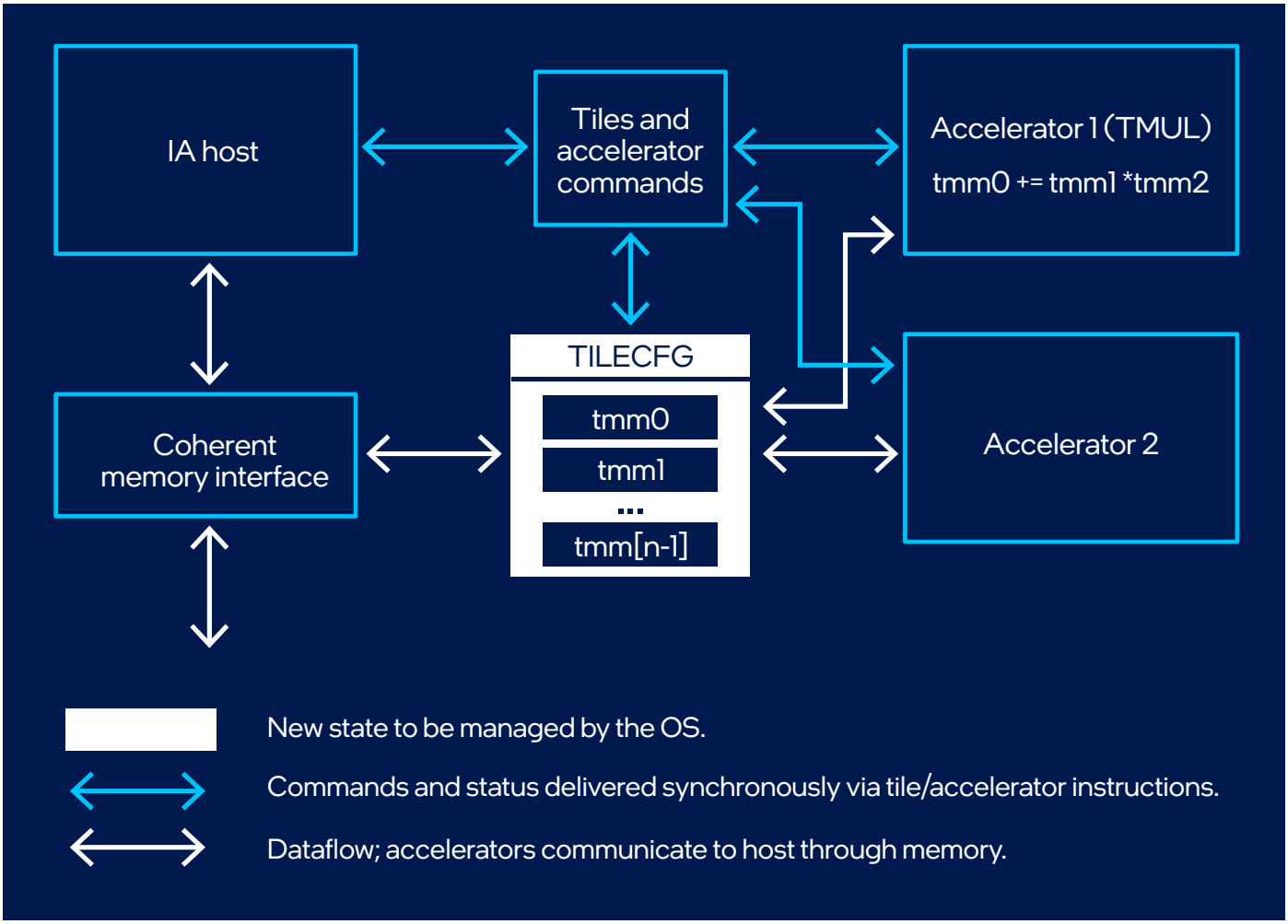

Figure 5. Intel AMX architecture

Performing a complete matrix multiplication (a "matrix multiply") is a very complex computation. Writing such complex code every time it's necessary to perform matrix multiply operations is not cost-effective.

Fortunately, oneDNN can help simplify this work. We only need call the MatMul primitive with some post-ops and pass several parameters (for example, m, n, k, stride, and data address). Just like its predecessor Intel Math Kernel Library (MKL), oneDNN will complete remaining work such as configuring the tile register files, loading data from memory, performing matrix multiply computation with post-ops, storing the result in memory, and releasing the tile register files. The use of Intel AMX is transparent to programmers via oneDNN, thus simplifying the programming required.

The entire flowchart of BERT running on a 4th Gen Intel Xeon Scalable processor with Intel AMX is shown in Figure 6.

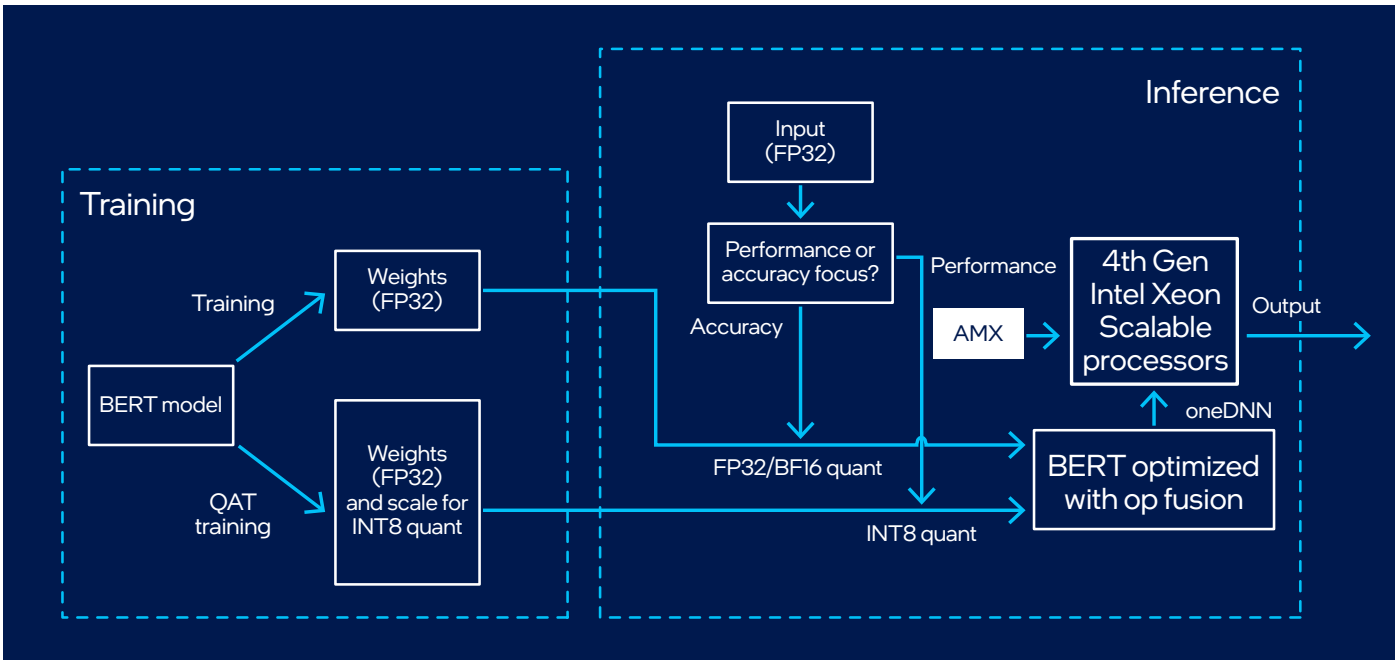

Figure 6. Entire flowchart of BERT with Intel AMX on 4th Gen Intel Xeon Scalable processors

Performance Comparison

Intel AMX acceleration can greatly improve BERT performance. To show the generation-to-generation performance advantage, we need to compare performance on various platforms. During the collaboration with Intel, Tencent StarLake Lab contributed greatly to the performance comparison and optimization work by employing their deep understanding of x86 micro-architecture and their experience with performance tuning. These contributions were invaluable in proving the performance of BERT technology on 3rd Gen Intel Xeon Scalable processors and 4th Gen Intel Xeon Scalable processors.

Multiple optimization instances were performed on one socket, and the latency of each instance was kept consistent. The performance results presented in Figure 7 show that the system performance using 4th Gen Intel Xeon Scalable processors with Intel AMX was significantly better on both INT8 and BF16—2.05x and 3.02x, respectively—compared to 3rd Gen Intel Xeon Scalable processors.[1,2]

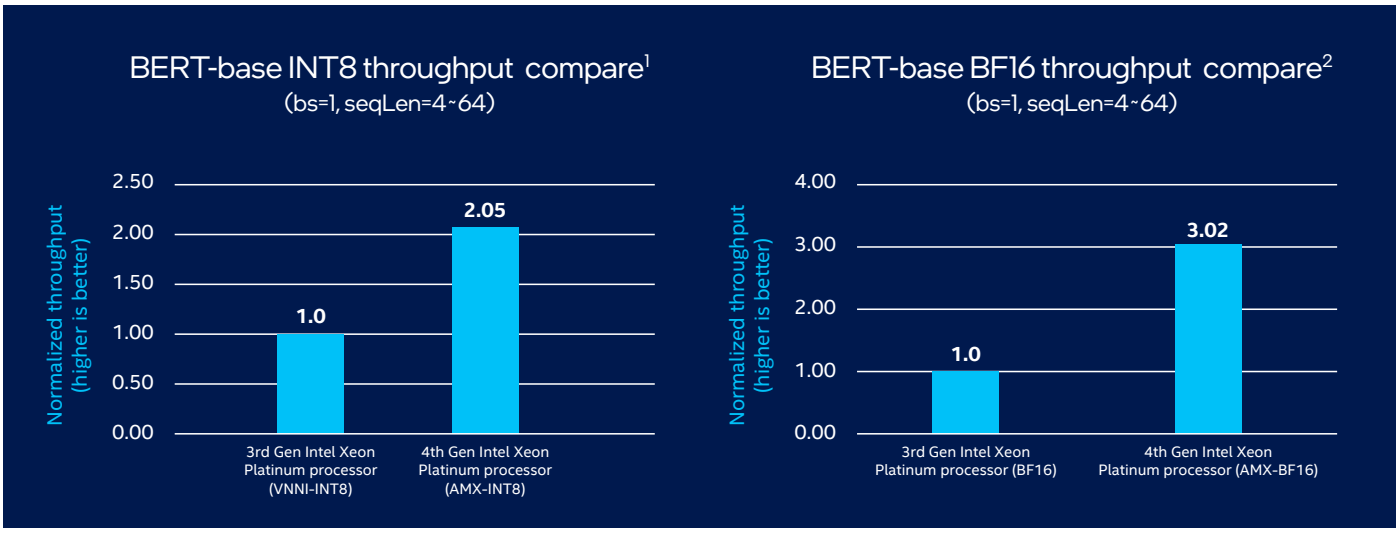

Figure 7. BERT model throughput gains using 4th Gen Intel Xeon Scalable processors with Intel AMX versus the previous generation

Summary

It's been demonstrated that 4th Gen Intel Xeon Scalable processors with Intel AMX can improve the performance of matrix multiply computations greatly through BF16/INT8 TMUL computing units and related instructions. Using Intel AMX, Intel and Tencent demonstrated BERT model throughput gains of 2x-3x versus the previous generation.[1,2] Now, Tencent can use the optimized BERT model to deliver better service experiences and to help reduce TCO.

intel XEON

[1] **BERT-base16 INT8 Throughput Comparison**

Baseline: Tencent TriRivers; BIOS version 1.08.00; OS CentOS Linux release 8.5.2111; kernel 4.18.0-348.7.1.el8_5.x86_64; microcode 0xd000375; IRQ balanced: Enabled; CPU Intel® Xeon® Platinum; Threads per core 2: Sockets 2: NUMA nodes 2; Prefetchers L2 HW, L2 Adj., DCU HW, DCU IP; Turbo Enabled; PPINs b59090a6f33f7966,b591426010edf86a; Power and Performance Policy: Performance; TDP 270 watts; Frequency driver intel_pstate; Frequency governor performance; Frequency (MHz) 2701; Max C-State 9; Installed memory 960GB (15x64GB DDR4 3200 MT/s [3200 MT/s])); Hugepagesize 2048 kB; Transparent Huge Pages always; Automatic NUMA balancing: Enabled; NIC 2x MT2892 Family [ConnectX-6 Dx], 1x device, 1x Ethernet interface; Driver summary 1x 111.8G INTEL SSDSCKHB12 Workload and version: BERT optimization for INT8; compiler GCC 8.5; libraries oneDNN-master-0721; Date tested 8/5/2022.

New: Intel Corporation ArcherCity; BIOS version EGSDCRB1.SYS.0090.D03.2210040200; OS CentOS Linux 8; kernel 5.16.0; microcode 0x2b0000c0; IRQ balanced: Enabled; CPU Intel® Xeon® Platinum; Threads per core 2; Sockets 2: NUMA nodes 2; Prefetchers L2 HW, L2 Adj., DCU HW, DCU IP; Turbo Enabled; PPINS 31461920530bed98,3143931fcf7f6036; Power and Performance Policy: Performance; TDP 350 watts; Frequency driver intel_pstate; Frequency governor performance; Frequency (MHz) 2494; Max C-State 9; Installed memory 512GB (16x32GB <OUT OF SPEC> 4800 MT/s [4800 MT/s]); Hugepagesize 2048 kB; Transparent Huge Pages always; Automatic NUMA balancing: Enabled; NIC 1x Ethernet Controller I225-LM, 1x Ethernet Controller E810-C for QSFP ; Driver summary 1x 349.3G INTEL SSDPE21K375GA, 1x 1.5T INTEL SSDPEDMD016T4, 1x 1.9T INTEL SSDPEKNW020T8 Workload and version: BERT optimization for INT8; compiler GCC 8.5; libraries oneDNN-master-0721; Date tested 10/19/2022.

[2] **BERT-base16 BF16 Throughput Comparison**

Baseline: Tencent QinghaiLake; BIOS version 1.02.00; OS Red Hat Enterprise Linux 8.2 (Ootpa); kernel 4.18.0-193.el8.x86_64; microcode 0x7002502; IRQ balance Enabled; CPU Intel® Xeon® Platinum; Threads per core 2; Sockets 4: NUMA nodes 4; Prefetchers L2 HW, L2 Adj., DCU HW, DCU IP; Turbo Enabled; PPINs 07ab90bc7f220116,07be7bc039fedc8f,07abcfbe92ff7f9b,07aba8bff01f278d; Power and Performance Policy: Performance; TDP 175 watts; Frequency driver acpi-cpufreq; Frequency governor performance; Frequency (MHz) 2695; Max C-State 9; Installed memory 1536GB (24x64GB DDR4 3200 MT/s [3200 MT/s]); Hugepagesize 2048 kB; Transparent Huge Pages always; Automatic NUMA balancing: Enabled; NIC 1x Ethernet interface; Driver summary: 1x 447.1G SSSTC ER2-GD480, 1x 894.3G Micron_5100_MTFDWorkload and version: BERT optimization for BF16; compiler GCC 8.3; libraries oneDNN-master-0721; Date tested 8/8/2022.

New: Intel Corporation ArcherCity; BIOS version EGSDCRB1.SYS.0090.D03.2210040200; OS CentOS Linux 8; kernel 5.16.0; microcode 0x2b0000c0; IRQ balanced: Enabled; CPU Intel® Xeon® Platinum; Threads per core 2; Sockets 2: NUMA nodes 2; Prefetchers L2 HW, L2 Adj., DCU HW, DCU IP; Turbo Enabled; PPINS 31461920530bed98,3143931fcf7f6036; Power and Performance Policy: Performance; TDP 350 watts; Frequency driver intel_pstate; Frequency governor performance; Frequency (MHz) 2552; Max C-State 9; Installed memory 512GB (16x32GB <OUT OF SPEC> 4800 MT/s [4800 MT/s]); Hugepagesize 2048 kB; Transparent Huge Pages always; Automatic NUMA balancing: Enabled; NIC 1x Ethernet Controller I225-LM, 1x Ethernet Controller E810-C for QSFP ; Driver summary 1x 349.3G INTEL SSDPE21K375GA, 1x 1.5T INTEL SSDPEDMD016T4, 1x 1.9T INTEL SSDPEKNW020T8 Workload and version: BERT optimization for BF16; compiler GCC 8.5; libraries oneDNN-master-0721; Date tested 10/19/2022.