intel XEON

白皮书

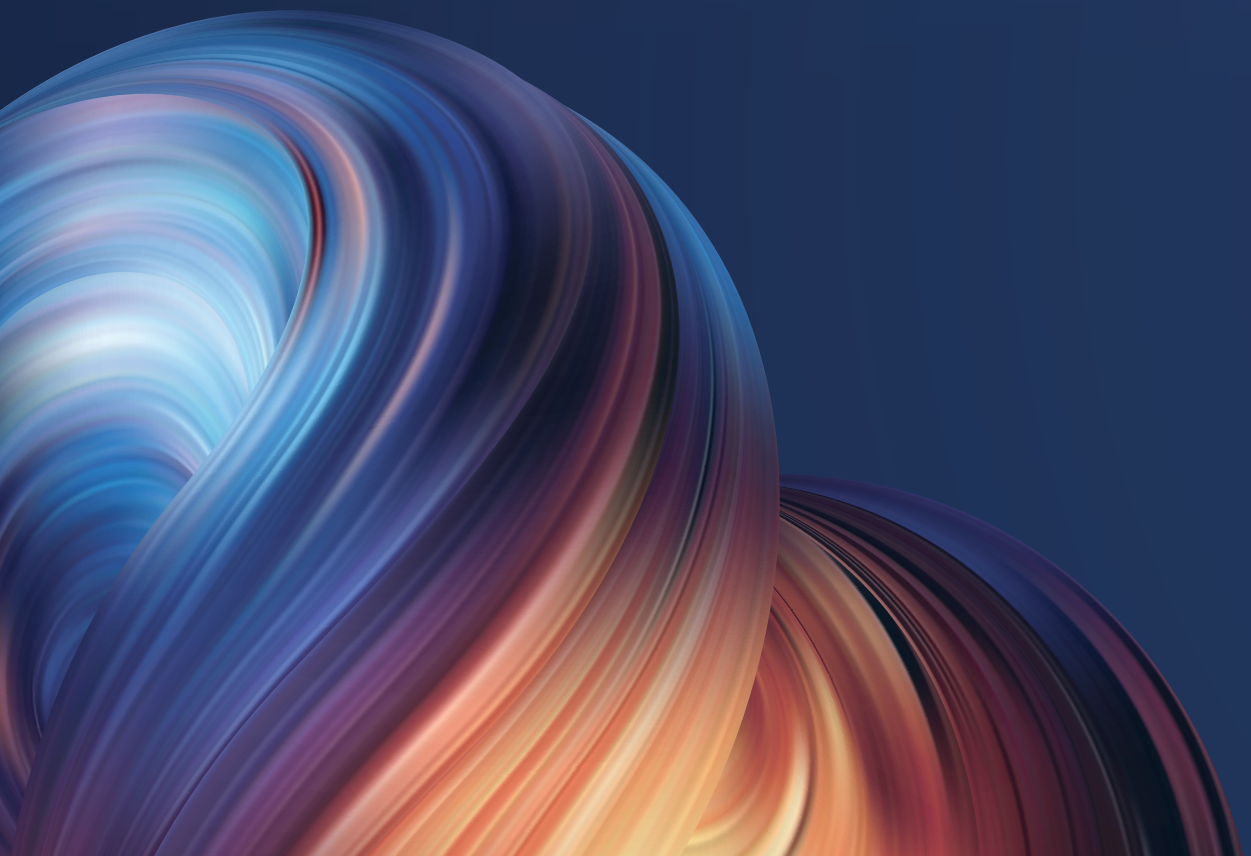
CPU 加速 AlphaFold2 更上一层楼！第四代至强® 可扩展平台带来 3.02 倍通量提升

AI for Science

AlphaFold2

第四代英特尔® 至强® 可扩展处理器

英特尔® oneAPI 工具套件



目录

概述

由 DeepMind 在 2021 年发布的 AlphaFold2，凭借自身在蛋白质结构预测上的高可信度，以及远优于传统实验方法的效率和成本表现，树起了一座“AI for Science”的全新里程碑。它不仅在生命科学领域掀起了颠覆式的革新，也成为了 AI 在生物学、医学和药学等领域落地的核心发力点。

随着 AlphaFold2 项目在产、学、研各细分领域中的启动与落地，其技术管线对于推理的高通量和高性能的需求也是与日剧增。一直活跃在“AI for Science”创新前沿的英特尔结合自身优势，以内置 AI 加速能力的产品技术，特别是第三代和目前最新的第四代至强® 可扩展平台为硬件基座，对 AlphaFold2 实施了端到端的高通量优化，并在实践中实现了比专用 AI 加速芯片更为出色的表现。其中，第三代英特尔® 至强® 可扩展处理器上的优化，可使通量提升至优化前的 23.11 倍[1]，而第四代英特尔® 至强® 可扩展处理器则可在此基础上使通量再获高达 3.02 倍的提升[2]。

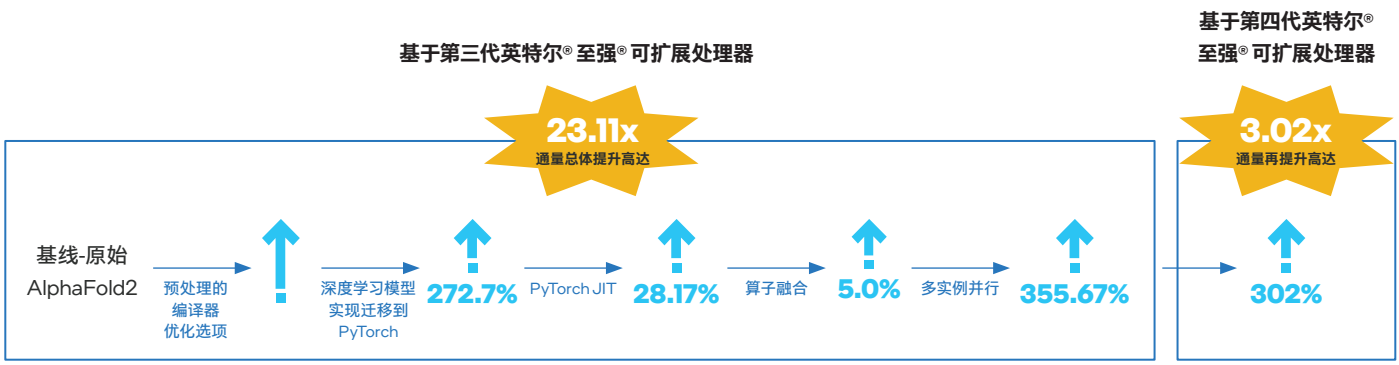
如此显著的优化成效，基于英特尔®架构的软硬件协作功不可没：

- **硬件支撑：** 英特尔® 至强® 可扩展平台的核心产品和技术特性，包括第三代和第四代英特尔® 至强® 可扩展处理器在算力输出上的越来越出色的表现，及其内置的 AI 加速引擎，如英特尔® 高级矢量扩展 512（英特尔® AVX-512）和英特尔® 高级矩阵扩展（Intel® Advanced Matrix Extensions，英特尔® AMX）等技术带来的预处理、推理计算优化，以及高带宽内存（High Bandwidth Memory, HBM）、全新 DDR5 内存等特性对张量吞吐、数据访存通量的明显提升；
- **软件加成：** 软件是充分利用或释放硬件加速潜能的“钥匙”，例如在模型推理阶段，序列长度为 n 的情况下，推理时间复杂度为 O（n2），此时原始 AlphaFold2 在 CPU 上的推理时长是难以接受的。英特尔为此采取了一系列软件调优举措，包括对注意力模块（attention unit）开展大张量切分（tensor slicing），以及使用英特尔® oneAPI 工具套件实施算子融合等优化方法，解决了 AlphaFold2 在 CPU 平台上面临的计算效率低和处理器利用率不足等难题，同时也缓解了调优方案执行各环节中面临的内存瓶颈等问题。

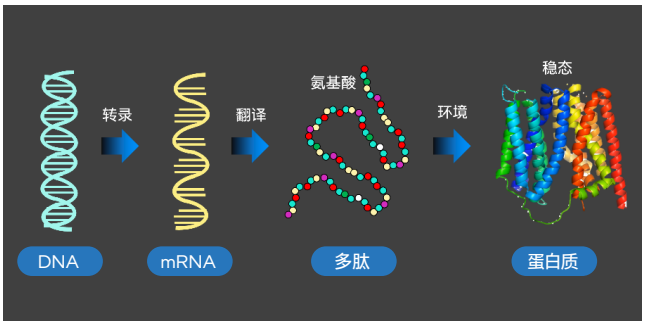
本文的核心任务，就是要介绍上述基于英特尔® 架构、致力于在 CPU 平台上加速 AI 应用的软硬件产品技术组合在 AlphaFold2 端到端优化中扮演的关键角色，并详细分享对它们进行配置、调优以求持续提升 AlphaFold2 应用性能表现的核心经验和技巧，从而为所有计划开展或正在推进类似探索、实践的合作伙伴及最终用户们提供一些关键的参考和建议，让整个产业界能够进一步加速相关应用的落地并尽可能提升其收益。

蛋白质结构解析任务繁重，AlphaFold2 勇担重任

如生物学中心法则（Central Dogma）所揭示的，脱氧核糖核酸（DNA）、核糖核酸（RNA）和蛋白质（包括多肽[4]）之间“转录 - 翻译”的关系，清晰呈现了有机体内的信息传递路径，也让人们认识到：对蛋白质三维结构开展有效解析与预测，就能对有机体的构成，及其运行和变化的规律实施更深层次的诠释和探究，进而可为生物学、医学、药学乃至农业、畜牧业等行业和领域的未来研究与发展提供高质量的生物学假设。



图一 基于英特尔® 至强® 可扩展平台的 AlphaFold2 推理优化路线图及其实现的性能提升[3]



图二 生物学中心法则

虽然许多基于传统实验方法的蛋白质结构解析工具，包括 X-射线晶体衍射、冷冻电镜、核磁共振等已获普遍运用，但通过传统实验方法进行结构解析的速度，远赶不上氨基酸序列的增加速度，这就造成海量待测样品 / 序列可能会在实验室中等待数月乃至数年才能得到解析。以 UniProtKB / Swiss-Prot 数据库搜集和整理的数据为例，单从实验获得的已知蛋白序列就已高达 57 万条之多[5]。

AI 技术的高速发展，则为破解上述效率问题提供了新的思路 - 人们开始将深度学习等方法运用于蛋白质结构预测，其中由 DeepMind 在 2020 年 CASP 14[6] 上提出的 AlphaFold2 方案尤其令人瞩目，它以惊人的 92.4 分 (GDT_TS 分数) 的表现实现了原子级别的预测精度，被认为“已可替代传统实验方法”[7]。

AlphaFold2 端到端预测：三个阶段协作增效

与以往多是间接预测蛋白质结构的 AI 方法不同，AlphaFold2 提供了完整的端到端蛋白质三维结构预测流程。如图三所示，其工作流程大致可分为预处理 (Preprocessing)、深度学习模型推理 (DL Model Inference) 以及后处理 (Postprocessing) 三个阶段，各阶段执行的功能如下：

- **预处理**

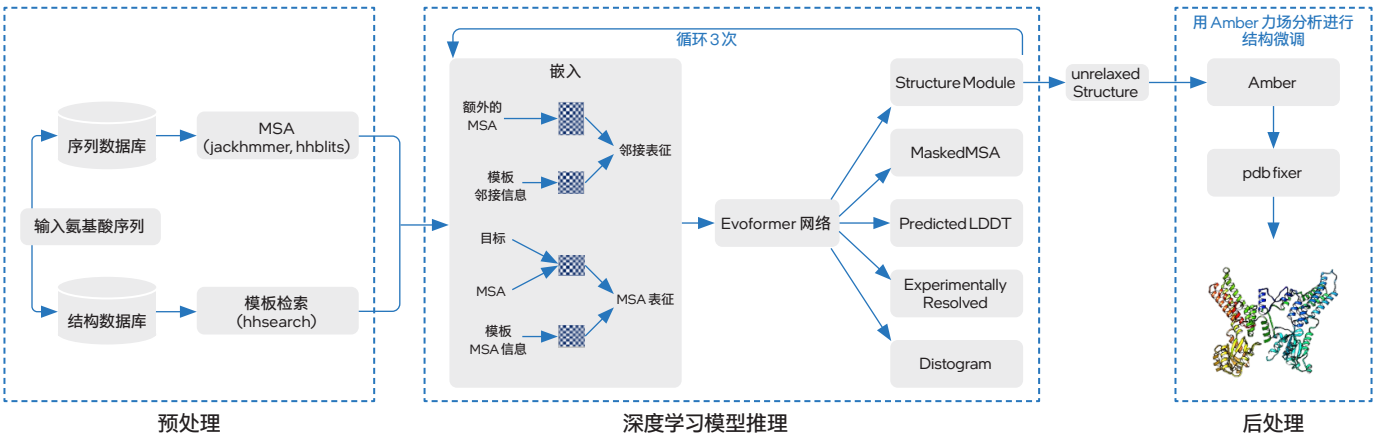
由于初始输入的氨基酸序列所含信息往往较少，因此 AlphaFold2 在预处理阶段会先利用已知信息 (包括蛋白质序列、结构模板) 来提升预测精度。包括借助一些蛋白质搜索工具在特定序列数据库中使用多序列比对 (MSA) 方法，以及在特定结构数据库中进行模板搜索，从而获得不同蛋白质之间的共有进化信息；

- **深度学习模型推理**

在该阶段中，AlphaFold2 首先会借助嵌入 (Embedding) 过程，将来自预处理阶段的模板 MSA 信息、MSA 和目标构成 MSA 表征 (MSA representation) 的三维张量，同时也将模板邻接信息和额外的 MSA 构成邻接表征 (pair representation) 的三维张量，随后两种表征信息会通过一个由 48 个块 (Block) 组成的 Evoformer 网络进行表征融合。在这一进程中，模型将通过一种 Self-Attention 机制来学习蛋白质的三角几何约束信息，并让两种表征信息相互影响来使模型推理出相应的三维结构，且循环三次；

- **后处理**

这一阶段，AlphaFold2 将使用 Amber 力场分析方法对获得的三维结构参数优化，并输出最终的蛋白质三维结构。



图三 AlphaFold2 基本架构

AlphaFold2 在预测精度上取得的优势，源于四点全新的设计思路：

- 在预处理阶段通过 MSA 方法等，将模板蛋白质结构和序列保守性信息融入预测特征；
- 在特征嵌入阶段，将保守性最高的 MSA 特征单独取出，压缩其余的 Extra MSA，并与模板特征交互；
- 在模型推理阶段，采用独特的双轨注意力模块和深层 Transformer 架构，并引入循环回收机制；
- 在结构网络层引入不变点注意力 (Invariant Point Attention) 机制。但这也意味着 AlphaFold2 从执行之初，直至整个推理过程都需要面对高通量的计算压力。

五大步骤：至强® 可扩展平台助 AlphaFold2 实现端到端优化

随着越来越多的科研机构、实验室和企业开始借助 AlphaFold2 进行蛋白质结构预测，各行业和领域内的使用者也开始遇到越来越多、也渐趋严峻的挑战。例如结构预测各环节面临着庞大的计算量，使用者需要更加充分地挖掘硬件的计算潜力来提升执行效率；为缩短结构预测时间，他们还需要利用更多计算节点来构建效率更高的并行计算方案等。

基于英特尔® 至强® 可扩展平台提供的内置 AI 加速能力，对于运算和存储性能的均衡设计，以及对硬件和软件协同优化能力的兼顾，英特尔着手对 AlphaFold2 进行了端到端的全面优化，以帮助生物学等应用领域的使用者们应对以上挑战。针对 AlphaFold2 的设计特点，该优化方案主要聚焦在预处理和模型推理两个层面，在第三代英特尔® 至强® 可扩展处理器和第四代英特尔® 至强® 可扩展处理器上，其基本可划分为以下五个步骤。

■ 第一步：预处理阶段 - 高通量优化

预处理阶段的高通量计算需求，使方案在执行时面临非常明显的并行计算压力。借助第三代或第四代英特尔® 至强® 可扩展处理器的多核优势及其内置的英特尔® AVX-512 技术，方案能够实现针对预处理阶段的高通量优化。

如前文所述，AlphaFold2 会在预处理阶段对特定序列数据库和结构数据库中的已知序列 / 模板信息进行搜索，包括使用 jackhmmer 等蛋白质搜索工具来执行 MSA 方法，即从数据库中抽取和输入与氨基酸序列相近的序列并进行对齐的过程，其目的是找出同源的序列 / 模板组成表征信息来为后续推理过程提供输入，由此提高预测精度。

这一过程中，计算平台需要执行大量的向量 / 矩阵运算。以模板搜索为例，其本质为计算两个隐马尔可夫模型 (Hidden Markov Model, HMM) 间的距离。当输入的氨基酸序列很长 (例如执行中输入长度达数百的氨基酸序列) 且需并行执行大量实例时，如果无法让处理器的算力“火力全开”去提升平台的并行计算效率，那么整个预处理过程的效率就会变得乏善可陈。

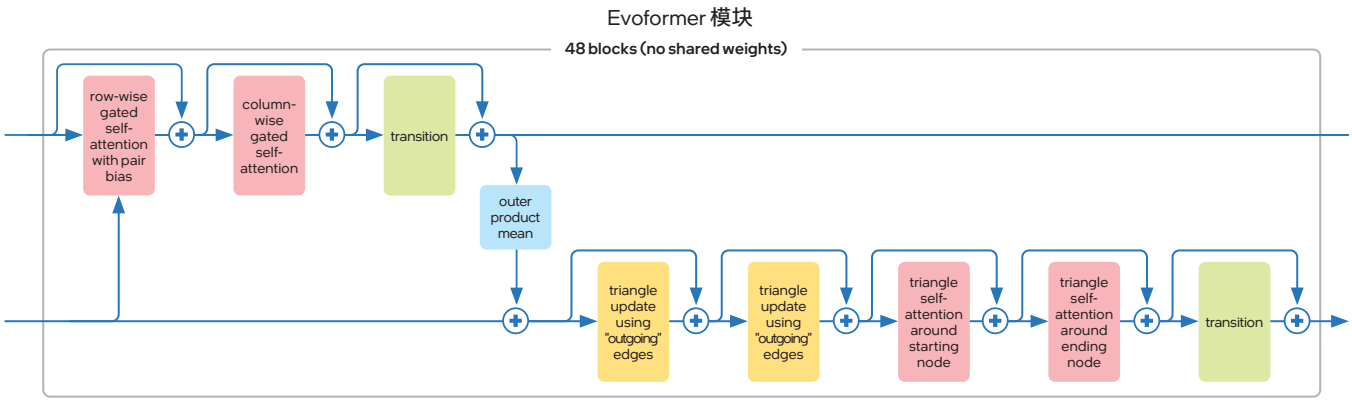
在实践中，由英特尔® 至强® 可扩展处理器内置的英特尔® AVX-512 以及使用英特尔® C++ 编译器配置 jemalloc 的优化方法，为方案提供了更进一步的性能调优空间。其中，针对序列 / 模板搜索所需的大量向量 / 矩阵运算需求，英特尔® AVX-512 技术，能以显著的高位宽优势 (最大可提供 512 位向量计算能力) 来提升计算过程中的向量化并行程度，从而有效提升向量 / 矩阵运算效率。

■ 第二步：模型推理阶段 - 将深度学习模型迁移至面向英特尔® 架构优化的 PyTorch

原始版本的 AlphaFold2 是基于 DeepMind 的 JAX 和 haiku-API 做的网络实现，但目前 JAX 上还没有面向英特尔® 架构平台的优化工具。而 PyTorch 拥有良好的动态图纠错方法，与 haiku-API 有着相似的风格，并可以采用面向 PyTorch 的英特尔® 扩展优化框架 (Intel® Extension for PyTorch，IPEX，可由英特尔® oneAPI AI 工具套件提供)。为实现更好的优化效果，方案选择将深度学习模型迁移至面向英特尔® 架构优化的 PyTorch，并最终逐模块地从 JAX / haiku 上完成了代码迁移。

■ 第三步：模型推理阶段 - PyTorch JIT

为提高模型的推理速度，便于利用 IPEX 的算子融合等加速手段，优化方案中还对迁移后的代码进行了一系列的 API 改造，在不改变网络拓扑的前提下，引入 PyTorch Just-In-Time (JIT) 图编译技术，将网络最终转化为静态图。



Name	Self CPU %	CPU total %
aten::einsum	0.07%	42.23%
aten::bmm	26.73%	37.51%
quantized::Linear_dynamic	20.15%	20.16%
aten::clone	0.08%	16.45%
aten::copy_	16.35%	16.35%
aten::add	12.62%	12.62%
aten::softmax	0.00%	8.76%
aten::_softmax	8.76%	8.76%

Large Tenors (~140MB per tensor)

图四 Evoformer 模块的热点算子

■ 第四步：模型推理阶段 - 切分 Attention 模块和算子融合

AlphaFold2 的嵌入过程是构成 MSA 表征张量和邻接表征张量来作为 Evoformer 网络输入的关键步骤。从其算法设计可以获知，其注意力模块中包含了大量的偏移量 (bias) 计算。

这种偏移量计算是通过张量间的矩阵运算来完成的，因此运算过程中会伴随张量的扩张。当张量达到一定规模后，扩张过程对内存容量的需求就会变得巨大。这就使 AlphaFold2 在嵌入过程中面临两个问题：一方面是巨大的内存峰值压力，其需求量会使内存资源在短时间耗尽，尤其是内存峰值在相互叠加之后，进而可能造成推理任务的失败；另一方面，大张量运算所需的海量内存也会带来不可忽略的内存分配过程，从而增加执行耗时。

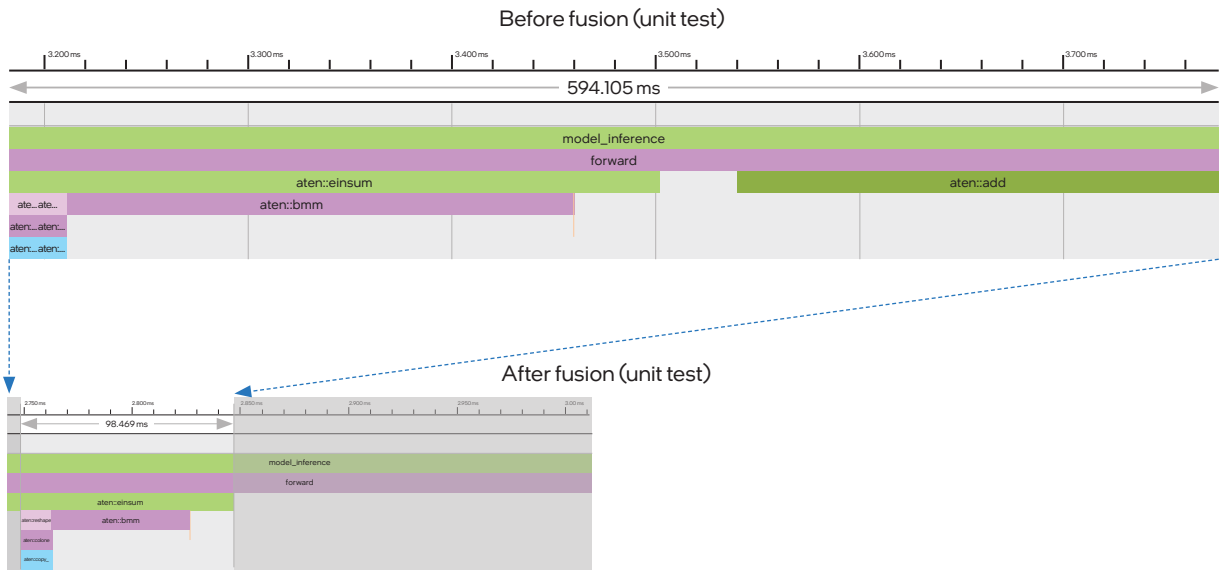
为此，英特尔提出了“对注意力模块进行大张量切分”的优化思路，即，将大张量切分为多个较小的张量，来降低扩张中的内存需求。例如将上述“5120 x 1 x 1 x 64”的张量切分为“320 x 1 x 1 x 64”后，其扩张所需的内存就由 930MB 降至 59.69MB，仅为未进行张量切分时的 6.4% 左右，有效消减了内存峰值压力。

同时，英特尔也发现利用 PyTorch 自带的 Profiler 对 AlphaFold2 的 Evoformer 网络进行算子跟踪分析时，Einsum 和 Add 这两种算子占用了大部分的算力资源。因此，英特尔就考虑使用 IPEX (建议版本为 IPEX-2.0.100 + cpu 或更高) 提供的算子融合能力来实现上述两种计算过程的融合。

传统的深度学习计算过程都是逐一操作：例如 Einsum 计算过程结束后，函数返回值需要在 Python 进程中建立一个临时缓存，然后通过调用 Add 算子，再次进入 oneDNN 完成第二个函数的运算，这中间来回折返的过程时间消耗不可忽略。如图五所示，算子融合带来的优势就在于，在前一操作结束后可以马上执行后一操作，节省了中间建立临时缓存数据结构的时间。同时从时间轴上不难看出，经过融合后，两个连续的算子合并为一个，用时也显著缩短。

■ 第五步：模型推理阶段 - 破解多实例运算过程中的计算和内存瓶颈

为了让推理性能在多实例进程中获得更接近线性的增长表现，优化方案也借助英特尔® 至强® 可扩展平台提供的高效且更为均衡的计算和存储优势实施了有针对性的优化。



图五 算子 Einsum + Add 融合效果图

这些优势包括借助基于 NUMA 架构的核心绑定技术，来充分挖掘至强® 可扩展处理器的多核心优势。这一技术可对处理器节点以及访问本地内存进程予以精确控制，让每个推理工作负载都能稳定地在同一组核心上执行，并优先访问对应的近端内存，从而提供更优、也更稳定的并行算力输出。

四剂“强芯针”：新一代至强® 可扩展处理器为 AlphaFold2 推理带来“推背感”

英特尔硬件与加速器的持续更新，正为 AlphaFold2 带来更大的优化空间。随着全新第四代英特尔® 至强® 可扩展处理器被逐步引入 AlphaFold2 的工作环境，其不仅提供了更强的基础算力，也带来多项针对 AI 工作负载的优化加速技术。在基于第四代英特尔® 至强® 可扩展处理器的优化工作中，英特尔基于以下四剂“强芯针”，让 AlphaFold2 的推理性能获得更为显著的提升。

1. 借助 TPP 技术，降低推理过程中的内存消耗

在深度学习系统开发中，诸如算子 (Operators)、算法概念 (Algorithmic Concepts) 以及计算模式 (Computational Motifs) 等编程范式 (Programming Paradigm) 通常会面向特定平台进行调优，这会对系统的构建便利性、性能调优以及可移植性造成障碍。为此，张量计算原语 (Tensor Processing Primitives，TPP) 技术是在 2D 张量上定义了一组低层级的基本算子，通过有效、且可移植的张量级算子来应对这一问题。TPP 可被看成是一种虚拟的张量指令集架构，能将英特尔® AVX-512 等物理指令集予以抽象，并生成经优化的平台代码。

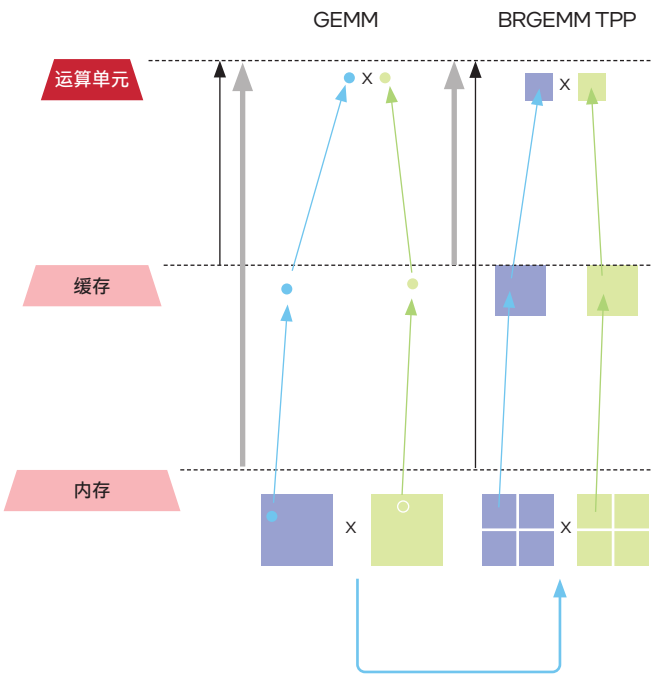
根据自身软硬件特性，英特尔面向 PyTorch 对 TPP 进行了扩展。面向 PyTorch 的英特尔® TPP 扩展 (Intel® Tensor Processing Primitives Extension for PyTorch) 不仅能让开发者直接使用 TPP 调用英特尔® oneAPI 等库来生成优化代码，也可利用面向 PyTorch 的 TPP 作为构建块来表示底层张量计算。

在优化方案中引入 TPP 技术能让 AlphaFold2 在通用矩阵乘法 (General Matrix Multiplication，GEMM) 等计算中获得优势，降低内存消耗并更好地利用全新第四代英特尔® 至强® 可扩展处理器所具备的大容量末级缓存优势。例如在 Evoformer 模块中需要进行大量的狭长矩阵乘法运算。对于在处理器上执行的矩阵乘法计算，一般会采用两种重要的优化方式：

- 以单指令多数据 (Single Instruction Multiple Data，SIMD) 方式处理数据；
- 优化内存访问模式，提升缓存命中率来提高数值计算和访存效率。

通过引入面向 PyTorch 的英特尔® TPP 扩展，英特尔在 AlphaFold2 实现了以上两种优化。如图六所示，一方面由 libxsmm

(小矩阵乘法函数库) 构建起来的 TPP 能借助 BRGEMM (Batch Reduce General Matrix Multiplication) 最大化利用第四代英特尔® 至强® 可扩展处理器所内置的 SIMD 运算单元，同时小矩阵乘法也能有效提高缓存命中率，使处理器的大容量末级缓存优势在计算过程中获得更充分的利用。TPP 的引入，令狭长矩阵乘法的空间复杂度从 O (n^2) 降为 O (n)，这使得运算过程中所需的内存峰值大幅降低，有效缓解了长序列蛋白质结构预测工作中面临的“序列长度天花板”问题。



图六 以 TPP 技术来充分利用新处理器的缓存优势

2. 支持 DDR5 内存与大容量缓存带来张量吞吐提升

通过对算法架构的解析可知，AlphaFold2 中大量的矩阵运算过程都需要内存予以支撑，因此内存性能是影响 AlphaFold2 性能的重要因素。而随着预测序列长度的增加，计算中所需的内存也会成倍增加，内存性能，尤其是内存带宽对系统整体性能的影响也会更为明显。

与此同时，更优的缓存策略也能让 AlphaFold2 进一步发挥潜能。由于张量间的矩阵运算会涉及大量的内存数据访存，而更靠近处理器运算单元末级缓存在延迟性能上比内存高出一个数量级。因此在复杂的矩阵运算中，更多的热数据通过末级缓存而非内存来访存，可以带来显著的性能提升。

第四代英特尔® 至强® 可扩展处理器对 DDR5 内存的支持，以及所具备的大容量末级缓存，为张量吞吐量的提升提供了更佳途径。新一代 DDR5 内存不仅频率更高、工作电压更低，还具有远超 DDR4 内存的带宽速度。与 DDR4 内存 25.6GBps (3,200MHz) 的带宽相比，DDR5 内存带宽达到了 38.4GBps (4,800MHz) 以上，提升幅度超过了 50%。同时，新处理器的末级缓存也由上一代的最高 60MB 提升至本代的最高 112.5MB，提升幅度达到了 87.5%[8]。性能更高的内存与容量更大的末级缓存，使 AlphaFold2 推理过程中关键的张量吞吐获得了显著提升。

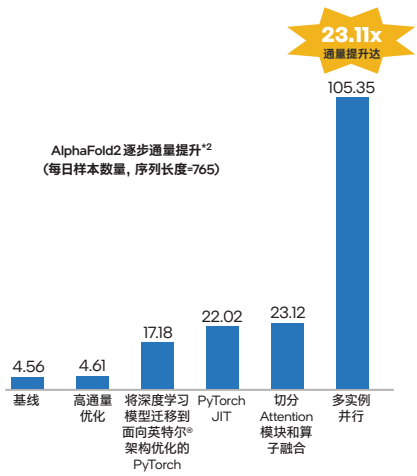
3. 全新英特尔® AMX_BF16 在保证精度的前提下加速推理过程

第四代英特尔® 至强® 可扩展处理器面向深度学习应用推出的“杀手锏”之一就是其创新的内置 AI 加速引擎，即英特尔® AMX。作为矩阵相关的加速器，英特尔® AMX 能显著加速基于 CPU 平台的深度学习推理和训练，提升 AI 整体性能。英特尔® AMX 对 INT8、BF16 等低精度数据类型都有着良好的支持 (通过 AMX_INT8、AMX_BF16 等不同指令集执行操作)，其中 BF16 数据类型在精度上有着不逊于 FP32 数据类型的表现。

针对 AlphaFold2 推理过程所需的大量矩阵运算，AMX_BF16 能在保持较高精度的同时，提高计算速度并减少存储空间。究其原因，是因为英特尔® AMX 在解决矩阵乘法问题时，直接采用了分块矩阵乘法的方式。其内部所定义的 Tile 矩阵乘法 (Tile Matrix Multiply Unit，TMUL) 加速模块，能直接对矩阵寄存器中的数据实施矩阵运算操作，由此运算效率可得到大幅提升。实践数据表明，AlphaFold2 在推理过程中使用 AMX_BF16 后，推理时间可缩短数倍之多。

4. 高带宽内存 HBM2e 增加访存通量

与第四代英特尔® 至强® 可扩展处理器一同发布、采用了相同微架构的英特尔® 至强® CPU Max 系列中，还加入了对 HBM 的支持，这也能让运行在其上的 AlphaFold2 推理负载更进一步。作为一种采用 3D 堆叠技术的全新内存技术，HBM 能为 AI 应用场景所需的各类计算负载提供更大的内存带宽支持。



优化路线图	管线通量增幅	英特尔架构亲和性	英特尔 AI 软件
原始 AlphaFold2	基线		
第一步：高通量优化		英特尔® AVX-512	英特尔® 编译器
第二步：将深度学习模型迁移到面向英特尔® 架构优化的 PyTorch	+272.7%	英特尔® AVX-512	面向英特尔® 架构优化的 Python 英特尔® MKL
第三步：PyTorch JIT	+28.17%	英特尔® AVX-512	PyTorch with MKL
第四步：切分 Attention 模块和算子融合	+5.0%	英特尔® AVX-512	Intel® Extension for PyTorch (IPEX)
第五步：多实例并行	+355.67%	英特尔® 傲腾™ 持久内存，提供 TB 级内存支持	

图七 基于第三代英特尔® 至强® 可扩展处理器的优化流程中多种优化措施带来的累计性能提升[10]

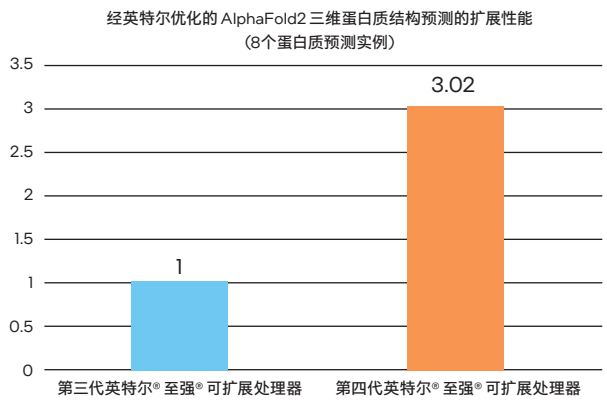
- 每个英特尔® 至强® CPU Max 系列都拥有 4 个基于第二代增强型高带宽内存 (HBM2e) 的堆栈，总容量为 64GB (每个堆栈的容量为 16GB)；
- 由于能同时访问多个 DRAM 芯片，因此 HBM 在带宽方面相较 DDR 技术更具优势，其中 HBM2e 可提供高达 1TB/s 的带宽；
- HBM 内存可根据工作负载特性，以“HBM Only”、“HBM Flat”以及“HBM Cache”三种不同的模式，通过灵活的配置与 DDR5 内存一起协同工作。

在实践中，HBM2e 内存被证明能有效缓解 AlphaFold2 推理负载对高带宽内存的需求，带来访存通量的大幅提升，从而降低整体的推理时长。

多个优化步骤实施后的总体性能表现

基于英特尔® 至强® 可扩展平台开展的 AlphaFold2 端到端优化，包括一系列并行计算能力优化举措的引入，使得整个 AlphaFold2 端到端处理过程的性能获得了质的提升。如图七所示，在基于第三代英特尔® 至强® 可扩展处理器的优化流程中，每个优化步骤获得的提升累积后，最后相比优化前通量提升可达 23.11 倍[9]。

而来自第四代英特尔® 至强® 可扩展处理器的优化加持，则使 AlphaFold2 的端到端通量获得再进一步的提升，如图八所示，与第三代英特尔® 至强® 可扩展处理器相比，融合 AMX_BF16、HBM 内存等技术的第四代英特尔® 至强® 可扩展平台能实现高达 3.02 倍的多实例通量提升[11]。



图八 第四代英特尔® 至强® 可扩展处理器带来多实例通量提升[12]

在探索和验证上述端到端 AlphaFold2 优化方案、步骤和经验的过程中，英特尔扮演的角色并非“独行侠”，而是与同在寻求相关解决方案的，专攻医药和生命科学研究和创新的产、学、研领域用户及合作伙伴们积极开展了广泛及深入的协作，这些协作起到了博采众长的效果，也为方案的普适性带来了助益。

同样，在优化方案基本定型，并展现了显著的通量提升效果以及能够担起更长序列蛋白质结构预测重任的能力后，众多合作伙伴与用户也第一时间参考和借鉴了方案中的方法、经验与技巧，并结合自身特定的环境、应用现状和需求，开展了实战验证和更进一步的探索。

总结与展望

得益于 AI 技术的高速发展和演进，它与科学前沿研究的结合正在快速地改变世界并造福人们的生活。始终走在 AI 应用创新与落地一线的英特尔，也在这一过程中借助至强 ® 可扩展平台，包括硬件层面的第三代英特尔® 至强® 可扩展处理器和第四代英特尔® 至强® 可扩展处理器，以及其软件层面的英特尔® oneAPI 工具套件等，基于这些软硬件之间的无缝组合与高效协作，以及多样化的 AI 优化方法，为 AlphaFold2 提供了持续改良的端到端高通量计算优化方案。

面向未来，英特尔还将继续携手科学前沿领域的合作伙伴，推进更多英特尔产品、技术与 AlphaFold2 等新技术开展交互与融合，在更多层面助力和加速“AI for Science”的技术创新，让 AI 应用为各类前沿科学研究和探索带来更多加速、助力与收获。

[1, 9, 10]测试配置:

- 测试组: 处理器: 2 x 英特尔® 至强® 铂金 8358 处理器, 内存: 16 x 32GB DDR4 3200MHz RDIMM + 16 x 256GB 英特尔® 傲腾™ 持久内存 200 系列 (Intel® Optane™ NMB1XXD256GPSU4 DCPMM), I/O 扩展: Raid Cntrlr - Trinity Dunes RAID Adapter,Intel RSP3TD160F, 存储: Solidigm Youngsville Refresh SSDSC2KB038T801 S4510 Series , 网络: SND I350-AM2 RJ45 Dual Port PCI-E4X_1KM, BIOS: Version: SE5C620.86B.01.01.0003.2104260124, Release Date: 04/26/2021, Linux 系统和 Kernel: Ubuntu 20.04 kernel-5.5.0-81-generic, Python 版本: 基于英特尔® 架构优化的 Python 3.9.7, AI 框架: PyTorch 1.11.0+cpu, Intel PyTorch Extension 1.11.100 with oneDNN 2.6, 其他工具和库: JAX 0.3.4 , JAXlib 0.3.2+cuda11.cudnn82 ,HMMER 3.3.2, HH-Suite 3.3.0, OpenMM 7.5.1;
- 对比组: 处理器: 2 x 英特尔® 至强® 铂金 8358 处理器, 内存: 32 x 128GB DDR4 3200MHz RDIMM , I/O 扩展: Raid Cntrlr - Trinity Dunes RAID Adapter, Intel RSP3TD160F , 存储: Solidigm Youngsville Refresh SSDSC2KB038T801 S4510 Series , 网络: SND I350-AM2 RJ45 Dual Port PCI-E4X_1KM, BIOS: Version: SE5C620.86B.01.01.0003.2104260124, Release Date: 04/26/2021, Linux系统和 Kernel: Ubuntu 20.04 kernel-5.5.0-81-generic, Python 版本: 基于英特尔® 架构优化的 Python 3.9.7, AI 框架: PyTorch 1.11.0+cpu, Intel PyTorch Extension 1.11.100 with oneDNN 2.6, 其他工具和库: JAX 0.3.4 , JAXlib 0.3.2+cuda11.cudnn82 ,HMMER 3.3.2, HH-Suite 3.3.0, OpenMM 7.5.1。

[2, 11, 12]测试配置:

-测试组: 处理器: 2 x 英特尔® 至强® CPU Max 系列 @ 1.90GHz, 内存: 128GB (8x16GB HBM2 3200MT/s, 存储: 1x 931.5G INTEL SSDPE2KX010T8 , 网络: 1x Ethernet Controller X710 for 10GBASE-T, BIOS: SE5C7411.86B.8424.D03.2208100444, Linux 系统和Kernel: CentOS Stream 8/5.19.0-rc6.0712.intel_next.1.x86_64+server, Python版本: 基于英特尔® 架构优化的Python 3.9.7, AI框架: PyTorch 1.11.0+cpu, Intel Extension for PyTorch 1.11.200 special branch for AlphaFold2, 其他工具和库: JAX 0.3.14 ;

-对比组: 处理器: 2x英特尔® 至强® 铂金 8360Y 处理器@2.40GHz, 内存: 512GB(16x32GB DDR4 3200MT/s) , 存储: 1x 894.3G INTEL SSDSC2KG96, 网络: 1x I210 Gigabit Network Connection, 2x Ethernet Controller 10G X550T, BIOS Version: WLYDCRB1.SYS.0021.P21.2106280839, Linux 系统和 Kernel: CentOS Linux 8/4.18.0-240.22.1.el8_3.x86_64, Python版本: 基于英特尔® 架构优化的 Python 3.9.7, AI 框架: PyTorch 1.11.0+cpu, Intel Extension for PyTorch 1.11.200 special branch for AlphaFold2, 其他工具和库: JAX 0.3.14。

[3]测试配置:

- 测试组: 处理器: 2 x 英特尔® 至强® 铂金 8358 处理器, 内存: 16 x 32GB DDR4 3200MHz RDIMM + 16 x 256GB 英特尔® 傲腾™ 持久内存 200 系列 (Intel® Optane™ NMB1XXD256GPSU4 DCPMM), I/O 扩展: Raid Cntrlr - Trinity Dunes RAID Adapter,Intel RSP3TD160F, 存储: Solidigm Youngsville Refresh SSDSC2KB038T801 S4510 Series , 网络: SND I350-AM2 RJ45 Dual Port PCI-E4X_1KM, BIOS: Version: SE5C620.86B.01.01.0003.2104260124, Release Date: 04/26/2021, Linux 系统和 Kernel: Ubuntu 20.04 kernel-5.5.0-81-generic, Python 版本: 基于英特尔® 架构优化的 Python 3.9.7, AI 框架: PyTorch 1.11.0+cpu, Intel PyTorch Extension 1.11.100 with oneDNN 2.6, 其他工具和库: JAX 0.3.4 , JAXlib 0.3.2+cuda11.cudnn82 ,HMMER 3.3.2, HH-Suite 3.3.0, OpenMM 7.5.1;
- 对比组: 处理器: 2 x 英特尔® 至强® 铂金 8358 处理器, 内存: 32 x 128GB DDR4 3200MHz RDIMM , I/O 扩展: Raid Cntrlr - Trinity Dunes RAID Adapter, Intel RSP3TD160F , 存储: Solidigm Youngsville Refresh SSDSC2KB038T801 S4510 Series , 网络: SND I350-AM2 RJ45 Dual Port PCI-E4X_1KM, BIOS: Version: SE5C620.86B.01.01.0003.2104260124, Release Date: 04/26/2021, Linux系统和 Kernel: Ubuntu 20.04 kernel-5.5.0-81-generic, Python 版本: 基于英特尔® 架构优化的 Python 3.9.7, AI 框架: PyTorch 1.11.0+cpu, Intel PyTorch Extension 1.11.100 with oneDNN 2.6, 其他工具和库: JAX 0.3.4 , JAXlib 0.3.2+cuda11.cudnn82 ,HMMER 3.3.2, HH-Suite 3.3.0, OpenMM 7.5.1。

测试配置:

-测试组: 处理器: 2 x 英特尔® 至强® CPU Max 系列 @ 1.90GHz, 内存: 128GB (8x16GB HBM2 3200MT/s, 存储: 1x 931.5G INTEL SSDPE2KX010T8 , 网络: 1x Ethernet Controller X710 for 10GBASE-T, BIOS: SE5C7411.86B.8424.D03.2208100444, Linux 系统和Kernel: CentOS Stream 8/5.19.0-rc6.0712.intel_next.1.x86_64+server, Python版本: 基于英特尔® 架构优化的Python 3.9.7, AI框架: PyTorch 1.11.0+cpu, Intel Extension for PyTorch 1.11.200 special branch for AlphaFold2, 其他工具和库: JAX 0.3.14 ;

-对比组: 处理器: 2x英特尔® 至强® 铂金 8360Y 处理器@2.40GHz, 内存: 512GB(16x32GB DDR4 3200MT/s) , 存储: 1x 894.3G INTEL SSDSC2KG96, 网络: 1x I210 Gigabit Network Connection, 2x Ethernet Controller 10G X550T, BIOS Version: WLYDCRB1.SYS.0021.P21.2106280839, Linux 系统和 Kernel: CentOS Linux 8/4.18.0-240.22.1.el8_3.x86_64, Python版本: 基于英特尔® 架构优化的 Python 3.9.7, AI 框架: PyTorch 1.11.0+cpu, Intel Extension for PyTorch 1.11.200 special branch for AlphaFold2, 其他工具和库: JAX 0.3.14。

[4]肽是 α- 氨基酸以肽键连接在一起而形成的化合物, 是蛋白质水解的中间产物, 由三个或三个以上氨基酸分子组成的肽叫多肽。

[5]数据援引自 UniProtKB/Swiss-Prot 数据库官网: https://web.expasy.org/docs/relnotes/relstat.html。

[6]CASP, 即结构预测的关键评估竞赛(Critical Assessment of Structure Prediction), 于 1994 年启动, 是对蛋白质结构的计算预测进行基准测试的一种手段。DeepMind 在 2020 年的 CASP14 上提出了 AlphaFold2 算法。

[7]一般认为, AI 方法的预测精度(GDT_TS 分数)超过 90 分, 可认为预测结果与实验方法得到的蛋白质结构基本一致。

[8]具体产品细节可参阅英特尔官网相关英特尔® 至强® 可扩展处理器产品介绍: https://www.intel.cn/content/www/cn/zh/products/details/processors/xeon/scalable.html

法律声明

英特尔并不控制或审计第三方数据。请您审查该内容, 咨询其他来源, 并确认提及数据是否准确。

性能测试结果基于 2022 年 5 月 9 日、2022年 9 月 16 日及 2022 年 9 月 19 日进行的测试, 且可能并未反映所有公开可用的安全更新。详情请参阅配置信息披露。没有任何产品或组件是绝对安全的。

英特尔技术特性和优势取决于系统配置, 并可能需要支持的硬件、软件或服务得以激活。产品性能会基于系统配置有所变化。没有任何产品或组件是绝对安全的。更多信息请从原始设备制造商或零售商处获得, 或请见 intel.com。

预测或模拟结果使用英特尔内部分析或架构模拟或建模, 该等结果仅供您参考。系统硬件、软件或配置中的任何差异将可能影响您的实际性能。

优化声明: 英特尔编译器针对英特尔微处理器的优化程度可能与针对非英特尔微处理器的优化程度不同。这些优化包括 SSE2、SSE3 和 SSSE3 指令集和其他优化。对于非英特尔微处理器上的任何优化是否存在、其功能或效力, 英特尔不做任何保证。

本产品中取决于微处理器的优化是针对英特尔微处理器。不具体针对英特尔微架构的特定优化为英特尔微处理器保留。请参考适用的产品用户与参考指南, 获取有关本声明中具体指令集的更多信息。

声明版本: #20110804

描述的产品可能包含可能导致产品与公布的技术规格有所偏差的、被称为非重要错误的设计瑕疵或错误。一经要求, 我们将提供当前描述的非重要错误。

声明: 本文仅用于宣传英特尔和合作伙伴的科技技术。英特尔不以任何方式宣传或介绍医疗机构、医疗服务, 也不为任何药品、医疗器械、保健食品等做推荐或证明。


intel