

通量提升达 23.11 倍！至强® 可扩展平台助力 AlphaFold2 端到端优化

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置信度，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。它不仅是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。

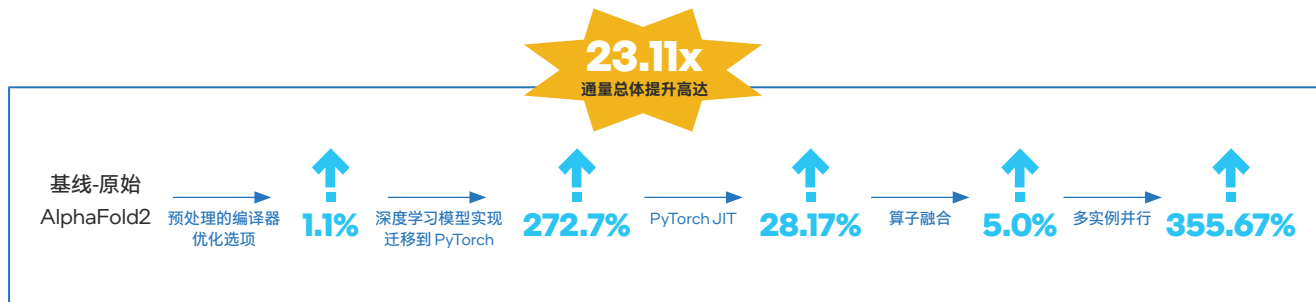
随着各类 AlphaFold2 项目在产、学、研各细分领域中的启动与落地，其技术管线对于推理的高通量和高性能的需求也是与日剧增。一直活跃在“AI for Science”创新前沿的英特尔结合自身优势，以内置 AI 加速能力的产品技术，特别是至强® 可扩展平台为硬件基座，对 AlphaFold2 实施了端到端的高通量优化，并在实践中实现了比专用 AI 加速芯片更为出色的表现——累计通量提升可达优化前的 23.11 倍¹。

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，例如第三代英特尔® 至强® 可扩展处理器在算力输出上的出色表现，及其内置的 AI 加速技术，如英特尔® 高级矢量扩展 512 (英特尔® AVX-512) 等技术带来的并行计算优化，还有英特尔® 傲腾™ 持久内存对“内存墙”障碍的突破，及这一突破对长序列高通量的并行推理优化的强力支持；
- **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 $O(n^2)$ ，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)

目录

概述	1
蛋白质结构解析任务繁重，AlphaFold2 生逢其时	2
AlphaFold2 端到端预测：三个阶段协作增效	3
五大步骤：至强® 可扩展平台助 AlphaFold2 实现端到端优化	4
第一步：预处理阶段-高通量优化	4
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	4
第三步：模型推理阶段-PyTorch JIT	5
第四步：模型推理阶段-切分 Attention 模块和算子融合	5
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	7
多个优化步骤实施后的总体性能表现	9
经验共享与实战验证	9
总结与展望	10



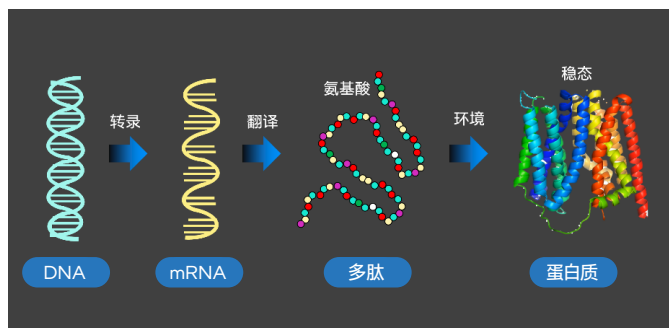
图一 基于英特尔® 至强® 可扩展平台的 AlphaFold2 推理优化路线图及其实现的性能提升²

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 生逢其时

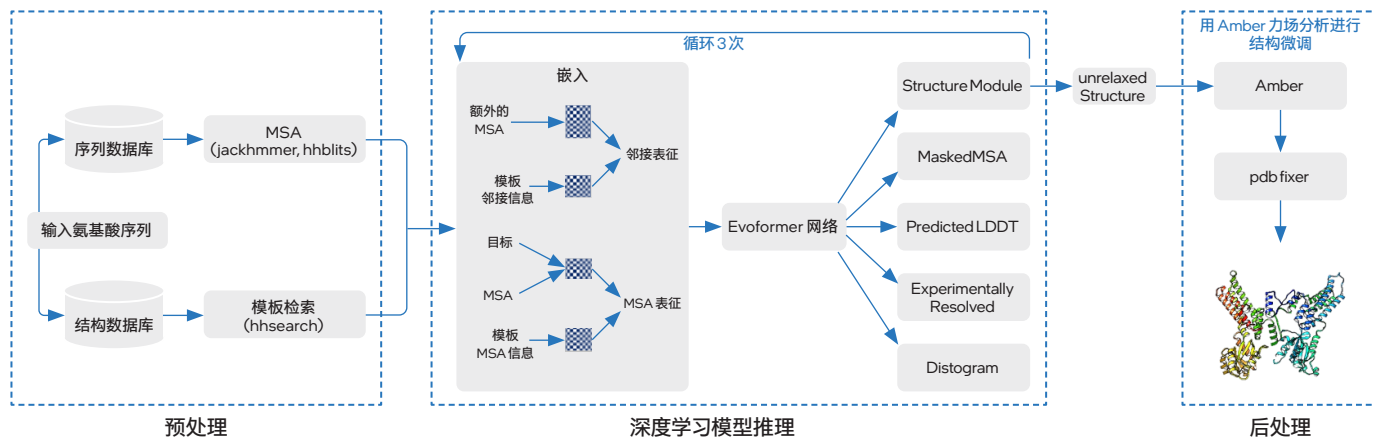
如生物学中心法则 (Central Dogma) 所揭示的, 脱氧核糖核酸 (DNA)、核糖核酸 (RNA) 和蛋白质 (包括多肽³、氨基酸) 之间“转录-翻译”的关系, 清晰呈现了有机体内的信息传递路径, 也让人们认识到: 对蛋白质三维结构开展有效解析与预测, 就能对有机体的构成, 及其运行和变化的规律实施更深层次的诠释和探究, 进而可为生物学、医学、药学乃至农业、畜牧业等行业和领域的未来研究与发展提供高质量的生物学假设。



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⁴。

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 14⁵ 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”⁶。



图三 AlphaFold2 基本架构

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。在这一过程中, 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。

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。

基于第三代英特尔® 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, 并可基本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, 第三代英特尔® 至强® 可扩展处理器一方面能凭借出色的微架构设计, 尤其是多核心、多线程和大容量高速缓存, 来保证 AlphaFold2 获得充足的总体算力, 以满足整个结构预测过程所需; 另一方面, 其内置的英特尔® AVX-512 及其支持的 NUMA (Non-Uniform Memory Access, 非一致存储访问) 架构等技术, 也为方案提供了更进一步的性能调优空间。

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。这一步在需要配备上述硬件平台的同时, 在 icc 编译器中做如下设置 (该设定支持所有英特尔® 至强® 可扩展处理器, 不仅限于代号为 Ice Lake 的第三代英特尔® 至强® 可扩展处理器):

```
1.-O3-no-prec-div-march=icelake-server
```

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® Extensions 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。

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。以一个“5120 x 1 x 1 x 64”的张量为例, 其初始内存需求为 1.25MB, 但在扩张过程中, 对内存容量的需求却可达 930MB。

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。相关代码示例如下:

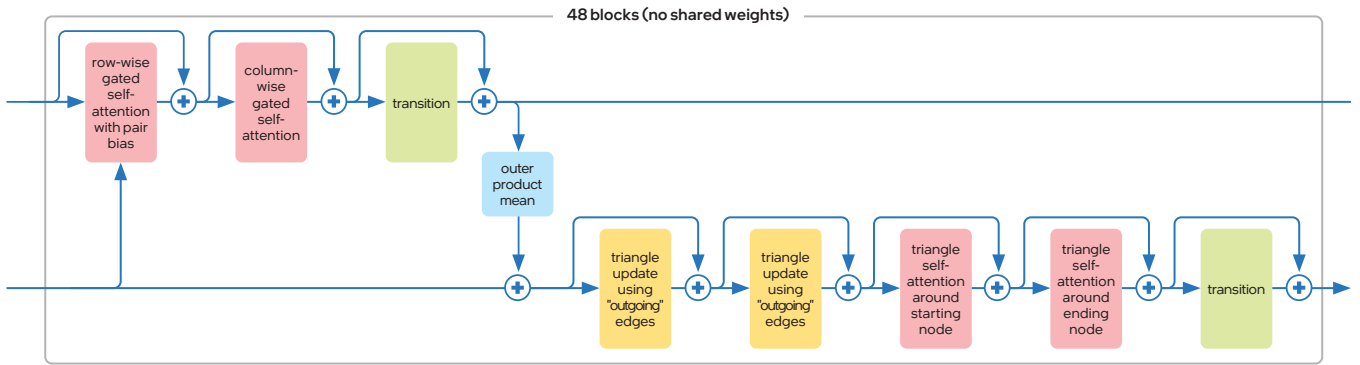
```

1. def _slice_attention(self, q_data, m_data, bias, nonbatched_bias):
2.     ###avoiding huge memory cost
3.     ##threshold is ajustable
4.     threshold = 1000
5.     unit 320 # unit is ajustable
6.     if q_data.size()[0] > threshold: #and is_stop_slice == 0:
7.         res = self.ones_like(q_data)
8.         for i in range(q_data.size()[0]//unit):
9.             q_sub_data = q_data[unit*i:unit*(i+1)]
10.            m_sub_data = m_data[unit*i:unit*(i+1)]
11.            bias_sub = bias[0:unit]
12.            res[unit*i:unit*(i+1)] =
self.attention(q_sub_data, m_sub_data, bias_sub, nonbatched_bias)
13.        return res
14.    else:
15.        return self.attention(q_data, m_data, bias, nonbatched_bias)

```

英特尔发现, 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-1.10.100 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。

Evoformer 模块

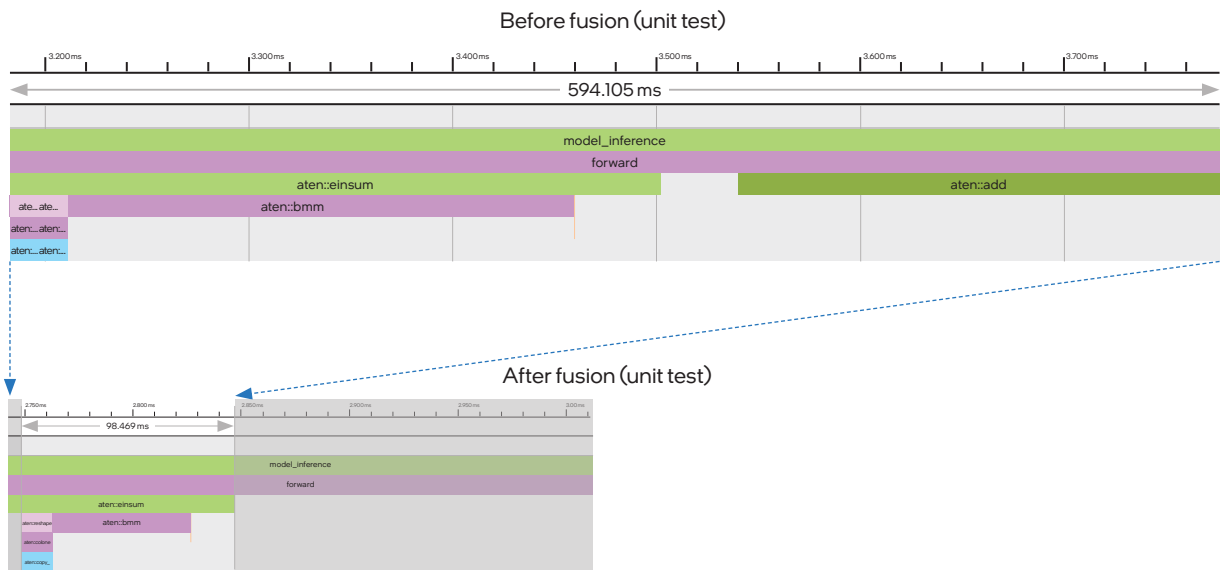


Name	Self CPU %	CPU total %
aten::einsum	0.07%	42.23%
aten::bmm	26.73%	37.51%
quantized:: Linear_dynamic	20.15%	20.16%
aten::clone	0.08%	16.45%
aten::copy_	16.35%	16.35%
aten::add	12.62%	12.62%
aten::softmax	0.00%	8.76%
aten::_softmax	8.76%	8.76%

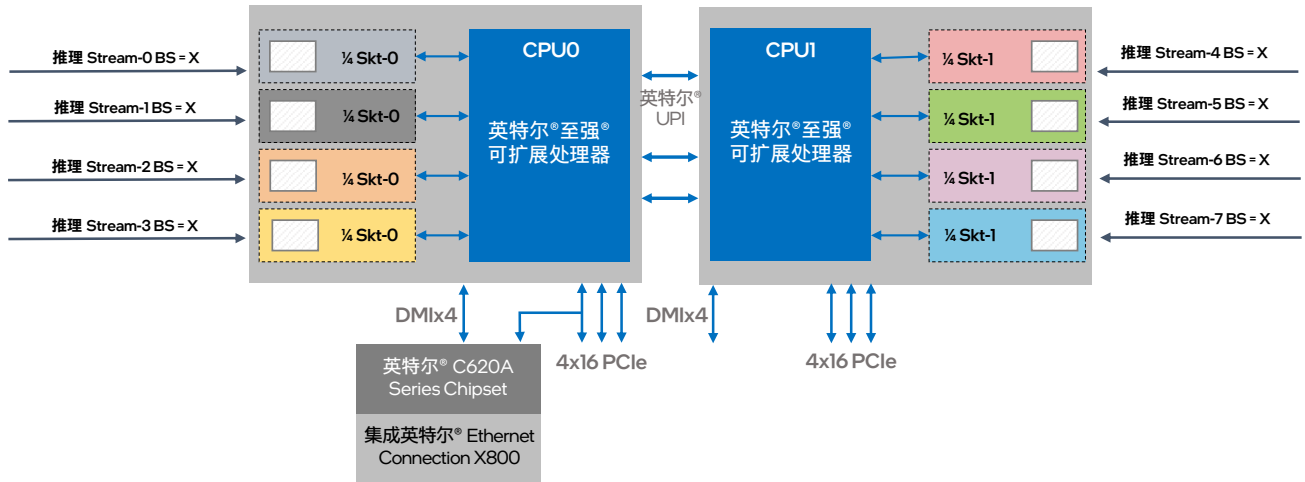
Large Tenors (~140MB per tensor)

图四 Evoformer 模块的热点算子

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。



图五 算子 Einsum+Add 融合效果图



图六 英特尔® 至强® 可扩展处理器提供多核并行算力输出

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。

方案首先是借助基于 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。如图六所示, 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。在执行中可使用以下 numactl 指令:

```
1. numactl -C $core_ids -m $socket_id $command
```

得益于英特尔® 至强® 可扩展处理器在微架构设计上的优势, 物理核与物理核之间的数据通信平均延时较短, 每个 NUMA 在并行计算中的工作效率也会更高。

同时, 在大规模服务器集群上开展多实例并行推理计算时, 节点间的数据交互量会呈平方增长, 导致大量占用通信带宽并损失计算效

率。英特尔® MPI 库的引入, 能针对并行计算的需求进行自动调整, 帮助方案实现更优的时延、带宽和可扩展性。方案中可以加入以下优化指令:

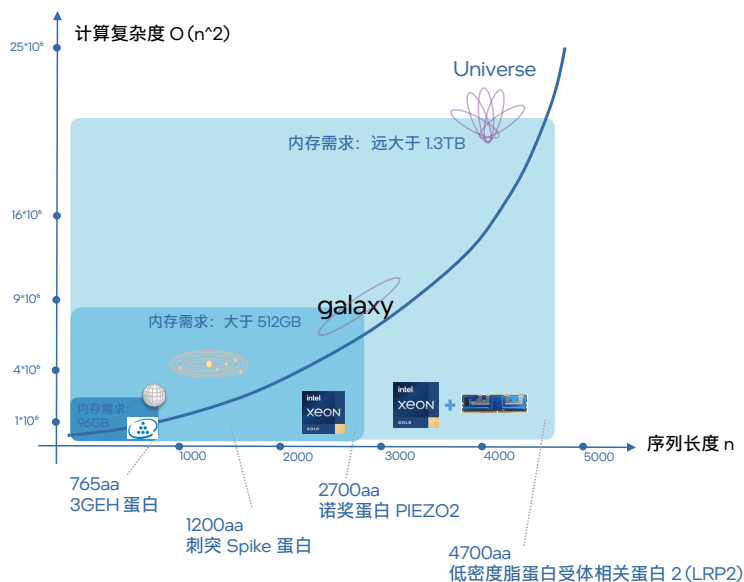
```
1. mpirun -np $total_instances --bind-to core:PE=1 --map-by socket -H $hostlist $cmd
```

在开展并行多实例计算优化之外, 英特尔还注意到, 内存的容量限制, 或者说瓶颈是限制 AlphaFold2 发挥潜能的另一个重要因素。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大容量内存予以支撑。其最大输入序列长度越长, 计算中所需的内存也就越大。而在并行计算能力得到有效优化后, 更多计算实例的加入也会进一步突显内存瓶颈问题。

受限于产品规格、主板架构和成本, 仅使用传统 DRAM (Dynamic Random Access Memory, 动态随机存取存储器) 内存很难实现 TB 级的大容量部署。英特尔® 傲腾™ 持久内存方案则是破解这一难题的有效途径, 基于创新的存储介质, 这一产品能为方案提供大容量和高性价比的内存支撑。

如图七所描绘的, 在面向不同蛋白质的结构预测工作中, 序列长度越长, 推理计算复杂度就越大。结合更多的并行计算, 所需的内存容量也就越高。如果用“星际探索”来比喻这种趋势, 那么:

- 对 3GEH 蛋白的结构预测就相当于探索地球;
- 而对某病毒相关的刺突 (Spike) 蛋白的结构预测就相当于将探索扩大到了整个太阳系;
- 对诺贝尔生理学或医学奖发现的 PIEZO2 蛋白结构进行预测则是进一步将探索扩展到了银河系;
- 对低密度脂蛋白受体相关蛋白 2 (LRP2) 的结构预测, 就好比是宇宙级的探索。



图七 大量长序列结构预测都会遇到“内存墙”问题

可见, 不同的探索范围, 所需耗费的资源(内存)也全然不同。在实践中, 进行 3GEH 蛋白(长度为 765aa)的结构预测, 内存大小在 100GB 就足以。而对 Spike 蛋白和 PIEZO2 蛋白进行预测时, 由于序列长度分别达到了 1200aa 和 2700aa, 就需要部署 512GB 范围的内存。而当人们对 LRP2 蛋白进行结构预测时, 其 4700aa 的序列长度要求的内存容量就远大于 1.3TB。如果 64 个实例并行执行, 内存容量的需求就会冲到一个令人惊叹的量级, 如果无法满足这个需求, 就会形成阻碍应用工作效能发挥的“内存墙”。

以英特尔® 傲腾™ 持久内存 200 系列为例, 其最高可提供 512GB 的单模组容量, 在与双路平台的第三代英特尔® 至强® 可扩展处理器搭配后, 在提供 3200MT/S 内存带宽的基础上, 理论上可实现每路高达 4TB 的英特尔® 傲腾™ 持久内存容量配置, 以及每路高达 6TB 的

内存总容量(与 DRAM 内存组合使用), 足以支撑 AlphaFold2 高密度部署方案。值得一提的是, 在提供更大容量的同时, 英特尔® 傲腾™ 持久内存还能输出接近 DRAM 内存的性能表现。

AlphaFold2 高通量推理实现所需的系统——英特尔推荐配置			
关键硬件配置		关键软件配置	
处理器	2 x 英特尔® 至强® 铂金 8358 处理器及更高型号	操作系统	CentOS 7.x/Ubuntu 20.04 及更高版本
DDR4 内存	16 x 32GB DDR4 3200MHz RDIMM	AI 框架	PyTorch 1.11.0 Intel PyTorch Extension 1.11.
英特尔® 傲腾™ 持久内存	16 x 256GB PMem (200 系列)	库和编译器	英特尔® oneAPI HPC 工具套件 2022.1.2 及更高版本
存储	英特尔® 傲腾™ 固态硬盘 3.8TB+	Python 版本	集成于英特尔® oneAPI HPC 工具套件的面向英特尔® 架构优化的 Python



图八 推理过程中多种优化措施带来的累计性能提升⁸

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和英特尔® 傲腾™ 持久内存产品的引入, 使得整个 AlphaFold2 端到端处理过程的性能获得了质的提升。如图八所示, 通过以上的优化流程, 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⁷。

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。以 AlphaFold2 为例, 虽然其问世时间不长, 但已经有生物学家将其应用到对抗新型传染病和其他疾病的研究中, 并取得了一定的成果⁹。

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, 包括其硬件层面的第三代英特尔® 至强® 可扩展处理器和英特尔® 傲腾™ 持久内存, 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 提供了端到端的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 + 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。

脚注和法律声明

1, 2, 7, 8 测试配置:

-测试组: 处理器: 2x 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 NMBIXXD256GPSU4 DCPMM), I/O 扩展: Raid Cntrlr - Trinity Dunes RAID Adapter, Intel RSP3TDI60F, 存储: Solidigm Youngsville Refresh SSDSC2KB038T801S4510 Series, 网络: SNDI350-AM2RJ45 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: 2x 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 RSP3TDI60F, 存储: Solidigm Youngsville Refresh SSDSC2KB038T801S4510 Series, 网络: SNDI350-AM2RJ45 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.

³ 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。

⁴ 数据援引自 UniProtKB/Swiss-Prot 数据库官网: <https://web.expasy.org/docs/relnotes/relstat.html>.

⁵ 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。

⁶ 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。

⁷ 信息源自 Deep learning guided optimization of human antibody against SARS-CoV-2 variants with broad neutralization: <https://www.pnas.org/doi/10.1073/pnas.2122954119>

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。

英特尔、英特尔标识以及其他英特尔商标是英特尔公司或其子公司在美国和/或其他国家的商标。

©英特尔公司版权所有