白皮书

# 英特尔助力构建开源大规模稀疏模型训练 / 预测引擎 DeepRec

DeepRec (PAI-TF) 是阿里巴巴集团统一的开源推荐引擎 (https://github.com/alibaba/DeepRec), 主要用于稀疏模型训练和预测, 可支撑千亿特征、万亿样本的超大规模稀疏训练, 在训练性能和效果方面均有明显优势; 目前 DeepRec 已支持淘宝搜索、推荐、广告等场景, 并广泛应用于淘宝、天猫、阿里妈妈、高德等业务。

英特尔自 2019 年以来就与阿里巴巴 PAI 团队紧密合作, 将英特尔人工智能 (Artificial Intelligence, AI) 技术应用到 DeepRec 中, 针对算子、子图、runtime、框架层和模型等多个层面进行优化, 以充分发挥英特尔软硬件优势, 助力阿里巴巴加速内外部 AI 业务性能。

## DeepRec 主要优势

当前主流的开源引擎对超大规模稀疏训练场景的支持尚有一定局限, 例如, 不支持在线训练、特征无法动态加载、线上部署迭代不方便等, 特别是性能难以达到业务需求的问题尤为明显。为解决上述问题, DeepRec 基于 TensorFlow1.15 针对稀疏模型场景进行了深度定制优化, 主要措施包含以下三类:

- **模型效果:** 主要通过增加 EmbeddingVariable (EV) 动态弹性特征功能以及改进 Adagrad Optimizer 来实现优化。EV 功能解决了原生 Variable size 大小难以预估、特征冲突等问题, 并提供了丰富的特征准入和淘汰策略等进阶功能; 同时, 针对特征出现频次进行冷热自动配置特征维度问题, 增加了高频特征表达力, 缓解了过拟合, 能够明显提高稀疏模型效果;
- **训练和推理性能:** 针对稀疏场景, DeepRec 在分布式、子图、算子、Runtime 等方面进行了深度性能优化, 包括分布式策略优化、自动流水线 SmartStage、自动图融合、Embedding 和 Attention 等图优化、常见稀疏算子优化、内存管理优化, 大幅降低了内存使用量, 显著加速了端到端的训练和推理性能;

## 目录

- **部署及 Serving:** DeepRec 支持增量模型导出和加载，实现了 10TB 级别的超大模型分钟级别的在线训练和更新上线，满足了业务对时效性的高要求；针对稀疏模型中特征存在冷热倾斜的特性，DeepRec 提供了多级混合存储（可达四级混合存储，即 HBM + DRAM + PMem + SSD）的能力，可在提升大模型性能的同时降低成本。

## 英特尔技术助力 DeepRec 实现高性能

英特尔与阿里巴巴 PAI 团队的紧密合作在实现以上三个独特优势中都发挥了重要作用，DeepRec 三大优势也充分体现了英特尔技术的巨大价值：

- 在性能优化方面，英特尔超大规模云软件团队与阿里巴巴紧密合作，针对 CPU 平台，从算子、子图、框架、runtime 等多个级别进行优化，充分利用英特尔® 至强® 可扩展处理器的各种新特征，更大程度发挥硬件优势；
- 为了提升 DeepRec 在 CPU 平台的易用性，还搭建了 modelzoo 来支持绝大部分主流推荐模型，并将 DeepRec 的独特 EV 功能应用到这些模型中，实现了开箱即用的用户体验。

同时，针对超大规模稀疏训练模型 EV 对存储和 KV 查找操作的特殊需求，英特尔傲腾创新中心团队提供基于英特尔® 傲腾™ 持久内存（简称“PMem”）的内存管理和存储方案，支持和配合 DeepRec 多级混合存储方案，满足了大内存和低成本需求；可编程解决方案事业部团队使用 FPGA 实现对 Embedding 的 KV 查找功能，大幅提升了 Embedding 查询能力，同时可释放更多的 CPU 资源。结合 CPU、PMem 和 FPGA 的不同硬件特点，从系统角度出发，针对不同需求更加充分地发挥英特尔软硬件优势，可加速 DeepRec 在阿里巴巴 AI 业务中的落地，并为整个稀疏场景的业务生态提供更优的解决方案。

### ■ 英特尔® DL Boost 为 DeepRec 提供关键性能加速

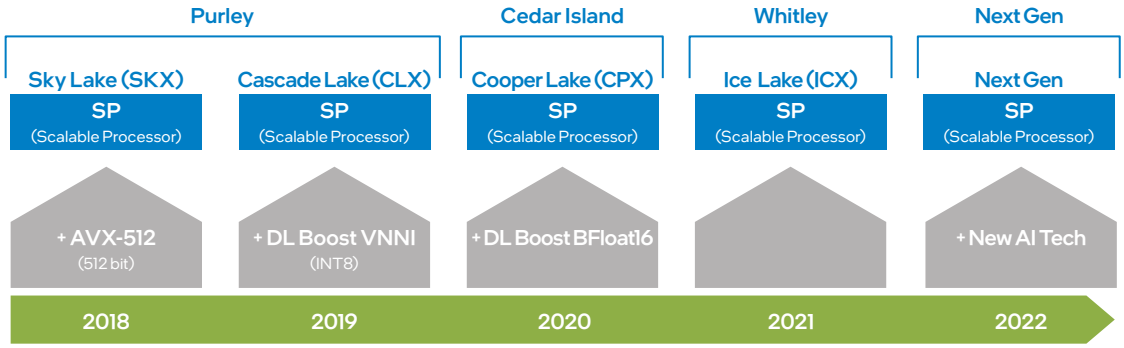
英特尔® DL Boost（英特尔® 深度学习加速）对 DeepRec 的优化，主要体现在**框架优化**、**算子优化**、**子图优化**和**模型优化**四个层面。

- **英特尔 x86 平台 AI 能力演进- 英特尔® DL Boost**

自英特尔® 至强® 可扩展处理器问世以来，通过从 AVX 256 升级到 AVX-512，英特尔将 AVX 的能力提高了一倍，极大地提升了深度学习训练和推理能力；而第二代英特尔® 至强® 可扩展处理器中又引入 DL Boost_VNNI，大幅提升了 INT8 乘加计算性能；自第三代英特尔® 至强® 可扩展处理器之后，英特尔推出支持 BFloat16（BF16）数据类型的指令集，来进一步提高深度学习训练和推理性能。随着硬件技术的不断创新和发展，英特尔将在下一代至强® 可扩展处理器推出新的 AI 处理技术，进一步提高 VNNI 和 BF16 从 1 维-向量到 2 维-矩阵的能力。上述的硬件指令集技术在 DeepRec 的优化中均已有所应用，使得针对不同的计算需求可使用不同的硬件特征，也验证了英特尔® AVX-512 和 BF16 非常适合稀疏场景的训练和推理加速。

- **框架优化**

DeepRec 集成了英特尔开源的跨平台深度学习性能加速 oneDNN（oneAPI Deep Neural Network Library），并且将 oneDNN 原有的线程池修改，统一成 DeepRec 的 Eigen 线程池，减少了线




图一 英特尔 x86 平台 AI 能力演进图


图二 Select 算子优化案例

程池切换开销，避免了不同线程池之间竞争而导致的性能下降问题。oneDNN 已经针对大量主流算子实现了性能优化，包括 MatMul、BiasAdd、LeakyReLU 等在稀疏场景中的常见算子，能够为搜广推模型提供强有力的性能支撑，并且 oneDNN 中的算子也支持 BF16 数据类型，与搭载 BF16 指令集的第三代英特尔® 至强® 可扩展处理器同时使用，可显著提升模型训练和推理性能。

在 DeepRec 编译选项中，只需加入“--config=mkl_threadpool”，便可轻松开启 oneDNN 优化。

- **算子优化**

oneDNN 虽可用来大幅提升计算密集型算子的性能，但搜索广告推荐模型中存在着大量稀疏算子，如 Select、DynamicStitch、Transpose、Tile、SparseSegmentMean 等，这些算子的原生实现大部分存在一定的访存优化空间，对此可采用针对性方案实现额外优化。该优化调用 AVX-512 指令，只需在编译命令中加入“--copt=-march=skylake-avx512”即可开启。以下为其中两个优化案例。

**案例一：** Select 算子实现原理是依据条件来做元素的选择，此时可采用英特尔® AVX-512 的 mask load 方式，如图二左图所示，以减少原先由 if 条件带来大量判断所导致的时间开销，然后再通过批量选择提升数据读写效率，最终线上测试表明，性能提升显著；

**案例二：** 同样，可以使用英特尔® AVX-512 的 unpack 和 shuffle 指令对 transpose 算子进行优化，即通过小 Block 的方式对矩阵进行转置，如图二右图所示，最终经线上测试表明，性能提升同样十分显著。

- **子图优化**

图优化是当前 AI 性能优化的主要有效手段之一。同样的，当 DeepRec 应用在大规模稀疏场景下时，通常存在着以 embedding 特征为主的大量特征信息处理，并且 embedding 中包含了大量小型算子；为了实现通用的性能提升，优化措施在 DeepRec 中加入了 fused_embedding_lookup 功能，对 embedding 子图进行融合，减少了大量冗余操作，同时配合以英特尔® AVX-512 指令加速计算，最终 embedding 子图性能提升显著。

通过在 tf.feature_column.embedding_column (..., do_fusion=True) API 将 do_fusion 设置为 True，即可开启 embedding 子图优化功能。

- **模型优化**

基于 CPU 平台，英特尔在 DeepRec 构建了涵盖 WDL、DeepFM、DLRM、DIEN、DIN、DSSM、BST、MMoE、DBMTL、ESMM 等多个主流模型的独有推荐模型集合，涉及召回、排序、多目标等多种常见的场景；并针对硬件平台进行性能优化，相较于其他框架，为这些模型基于 Criteo 等开源数据集在 CPU 平台上带来极大的性能提升。

其中表现最突出的当属混合精度的 BF16 和 Float32 的优化实现。通过在 DeepRec 中增加自定义控制 DNN 层数据类型的功能，来满足

稀疏场景高性能和高精度的需求；开启优化的方式如图三所示，通过 keep_weights 保留当前 variable 的数据类型为 Float32，用于防止梯度累加导致的精度下降，而后再采用两个 cast 操作将 DNN 操作转换成 BF16 进行运算，依托第三代英特尔® 至强® 可扩展处理器所具备的 BF16 硬件运算单元，极大地提升 DNN 运算性能，同时通过图融合 cast 操作进一步提升性能。

```
# DNN Layers
dnn_scope = tf.variable_scope('dnn_layers')
with dnn_scope.keep_weights(dtype=tf.float32) if ENABLE_BF16 else dnn_scope:
  if ENABLE_BF16:
    net = tf.cast(net, dtype=tf.bfloat16)

  net = self.dnn(net, self.dnn_hidden_units, "hiddenlayer") # BF16 datatype

  if ENABLE_BF16:
    net = tf.cast(net, dtype=tf.float32)
```

图三 混合精度优化开启方式

为了能够展示 BF16 对模型精度 AUC (Area Under Curve) 和性能 Gsteps/s 的影响，针对现有 modelzoo 的模型都应用以上混合精度优化方式。阿里巴巴 PAI 团队使用 DeepRec 在阿里云平台的评测表明[1]，基于 Criteo 数据集，使用 BF16 优化后，模型 WDL 精度或 AUC 可以逼近 FP32，并且 BF16 模型的训练性能提升达 1.4 倍，效果显著。

未来，为了更大程度地发挥 CPU 平台硬件优势，尤其是将新硬件特征的效果最大化，DeepRec 将从不同角度进一步实施优化，包括优化器算子、attention 子图、添加多目标模型等，以便为稀疏场景打造更高性能的 CPU 解决方案。

### ■ 使用 PMem 实现 Embedding 存储

对于超大规模稀疏模型训练和预测引擎（千亿特征、万亿样本、模型 10TB 级别），若全部采用动态随机存取存储器 (Dynamic Random Access Memory，DRAM) 来存储，会大幅提升总拥有成本 (Total Cost of Ownership，TCO)，同时给企业的 IT 运维和管理带来巨大压力，让 AI 解决方案的落地遭遇挑战。

PMem 具有更高存储密度和数据持久化优势，I/O 性能接近 DRAM，成本更为经济实惠，可充分满足超大规模稀疏训练和预测在高性能和大容量两方面的需求。

PMem 支持两种操作模式，即内存模式 (Memory Mode) 和应用直接访问模式 (App Direct Mode)。在内存模式中，它与普通的易失性（非持久性）系统存储器完全一样，但成本更低，能在保持系统预算的同时实现更高容量，并在单台服务器中提供 TB 级别的内存总容量；相比于内存模式，应用直接访问模式则可以利用 PMem 的持久化特性。在应用直接访问模式下，PMem 和与其相邻的 DRAM 内存都会被识别为可按字节寻址的内存，操作系统可以将 PMem 硬件作为两种不同的设备来使用，一种是 FSDAX 模式，PMem 被配置成块设备，用户可以将其格式化成一个文件系统来使用；另一种是 DEVDAX 模式，PMem 被驱动为单个字符设备，依赖内核 (5.1 以上) 提供的 KMEM DAX 特性，把 PMem 作为易失性内存使用，接入内存管理系统，作为一个和 DRAM 类似的、较慢较大的内存 NUMA节点，应用可透明访问。

在超大规模特征训练中，Embedding 变量存储占用 90% 以上的内存，内存容量会成为其瓶颈之一。将 EV 存到 PMem 可以打破这一瓶颈，创造多项价值，例如提高大规模分布式训练的内存存储能力、支持更大模型的训练和预测、减少多台机器之间的通信、提升模型训练性能，同时降低 TCO。

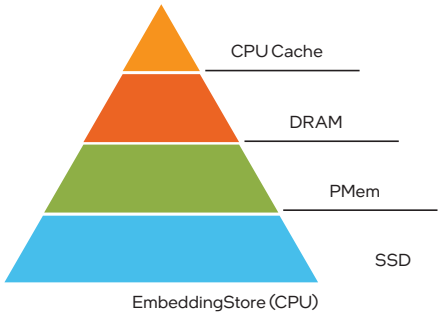
在 Embedding 多级混合存储中，PMem 同样是打破 DRAM 瓶颈的极佳选择。目前将 EV 存到 PMem 已具备三种方式，且在如下这三种方式下运行 micro-benchmark、WDL 模型和 WDL-proxy 模型，性能非常接近于将 EV 存到 DRAM，这无疑使得其 TCO 获得了很大优势：

- 将 PMem 配置成内存模式来保存 EV；
- 将 PMem 配置成应用直接访问 FSDAX 模式，并采用基于 Libpmem 库的分配器来保存 EV；
- 将 PMem 配置成 NUMA 节点并采用基于 Memkind 库的分配器来保存 EV。

阿里巴巴 PAI 团队在阿里云内存增强型实例 ecs.re7p.16xlarge 上采用 3 种保存 EV 的方式进行了 Modelzoo 中的 WDL 单机模型对比测试[2]，这 3 种方式分别是将 EV 存到 DRAM，采用基于 Libpmem

库的分配器来保存 EV 和采用基于 Memkind 库的分配器来保存 EV，测试结果表明将 EV 存到 PMem 与将 EV 存到 DRAM 的性能非常接近。

| | DRAM | PMem AD Mode FSDAX | PMem AD Mode KMEM DAX |
|---|---|---|---|
| Speedup | 1.0x | 0.95x | 0.96x |




图四 Embedding 多级混合存储

由此，下一步优化计划将采用 PMem 保存模型，把稀疏模型 checkpoint 文件存到持久内存中，来实现多个数量级的性能提升，摆脱目前用 SSD 保存恢复超大模型需要较长时间，且期间训练预测会中断的窘境。

## FPGA 加速 Embedding Lookup

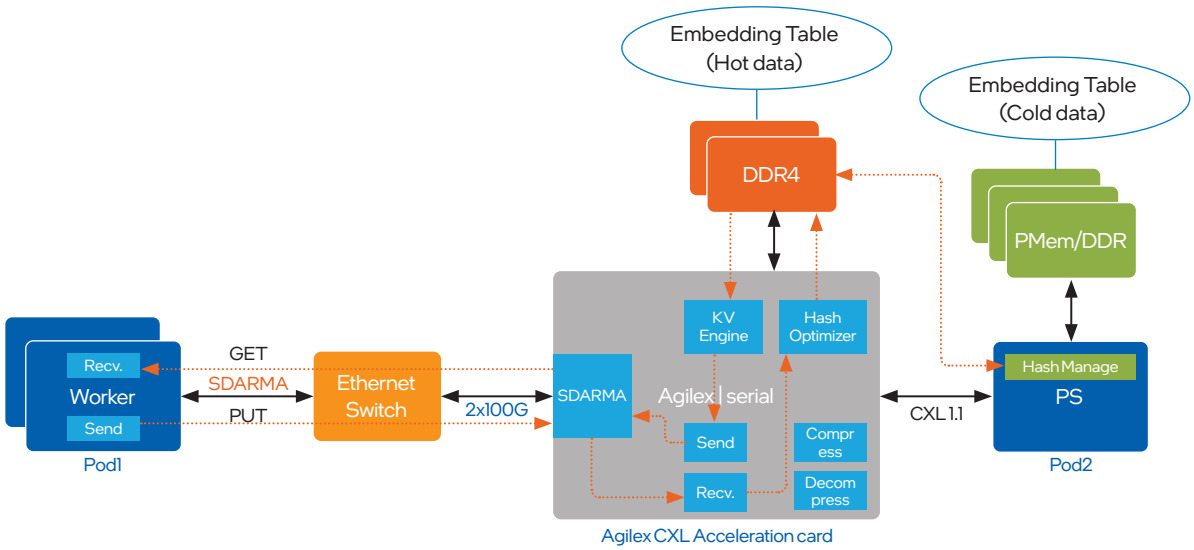
大规模稀疏训练及预测涵盖多种场景，例如分布式训练、单机和分布式预测以及异构计算训练等。它们与传统卷积神经网络 (Convolutional Neural Network，CNN) 或循环神经网络 (Recurrent Neural Networks，RNN) 相比有一个关键的不同，那就是 embedding table 的处理，而这些场景中的 Embedding table 处理需求面临新的挑战：

- 巨大的存储容量要求 (可达 10TB 或更多)；
- 相对低的计算密度；
- 不规则的 memory 访问模式。

DeepRec 通过 PS-worker 架构来支持超大规模任务场景。在 PS-worker 架构中，存储与计算分离，Embedding table 以 Key-Value 的形式被存储在 (几十、上百个) Parameter Servers 中，这些 PS 为 (几百、上千个) Worker 提供存取、更新模型参数的服务，其关键的指标就是流通量和访问时延。而面对大规模稀疏模型训练和预测，现有框架中 PS-worker 的实现就显露了其瓶颈：

- 用软件通过多线程方式实现的 KV engine 成为了流通量的瓶颈；
- 基于 TCP/RDMA 实现的 rpc 带来的开销，使得 Parameter Server 在分布式扩展时成为明显的时延和性能瓶颈。

为了解决流通量瓶颈和时延的问题，优化中引入了支持 CXL (Compute Express Link) 的英特尔® Agilex™ I 系列 FPGA，实施路径如图五所示。




图五 引入英特尔® Agilex™ I 系列 FPGA 实施优化

- 通过 FPGA 实现硬件的 KV engine 可以饱和内存或网络带宽，解决流通量瓶颈问题；
- 通过自定义支持可靠传输的 transport layer 协议，在同一个 FPGA 中处理 KV engine 和网络协议，不经过主机 CPU 直接根据 key 处理 value，以极低的时延和极小的抖动，消除 Parameter Server 在分布式扩展时的时延和性能瓶颈；
- 通过 CXL 提供的 cache-coherent 连接支持 HDM（Host Managed Device Memory）访问，设备端（FPGA卡）上采用 DDR4 以支持热数据访问的高性能需求，主机端使用 PMem 支持冷数据的存储，极大化降低 TCO；
- 以 FPGA 可以进一步实现 embedding table 的 in-line 处理，例如 tensor 操作，或者实现压缩及解压缩在网络带宽限制方面的突破。

基于英特尔® Agilex™ I 系列 FPGA 的加速方案能在一个硬件平台支持上述所有场景，流通量显著提升，同时提供较低的访问时延。

## 总结

前文介绍了针对 DeepRec 在 CPU、PMem 和 FPGA 不同硬件的优化实现方案，并成功部署到阿里巴巴多个内部和外部业务场景，在实际业务中也获得了明显的端到端性能加速，从不同角度解决了超大规模稀疏场景面临的问题和挑战。众所周知，英特尔为 AI 应用提供了多样化的硬件选择，为客户选择更优性价比的 AI 方案提供了可能；与此同时，英特尔与阿里巴巴及广大客户正一同基于多样化硬件实施软硬一体的创新协作和优化，从而更充分地发挥英特尔技术和平台的价值。英特尔也期望继续和业界伙伴合作展开更深入地合作，持续为 AI 技术的部署落地贡献力量。

intel

**脚注和法律声明**

[1] 如欲了解更多性能测试详情，请访问 https://github.com/alibaba/DeepRec/tree/main/modelzoo/WDL
[2] 如欲了解更多性能测试详情，请访问 https://help.aliyun.com/document_detail/25378.html?spm=5176.20205201010.0.0.787c4df5FgibRE#re7p

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 4 月 27 日和 2022 年 5 月 23 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。
描述的成本降低情景均旨在在特定情况和配置中举例说明特定英特尔产品如何影响未来成本并提供成本节约。情况均不同。英特尔不保证任何成本或成本降低。
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。