SIGIR19年有关GPU加速的一篇论文

解决的问题

FEXIPRO[SIGMOD'17]中的对IPR问题的求解较慢，可以使用GPU进行并行加速。

IPR问题

给定一个用户矩阵$Q \in \mathbb{R}^{d \times m}$以及一个物品矩阵$P \in \mathbb{R}^{d \times n}$，对于$Q$中的每一个用户$q$，返回内积$q^T P$中的前k个$q^T p$对应的物品列表$p$

做法及创新

行文逻辑

作者首先画出四个数据集上，SeqScan与FEXIPRO中两个步骤（内积计算与Top-k物品获取）的运行时间占比，发现内积计算占了总开销的90%以上，促使他提出方法加速这一步骤。接下来介绍GPU加速CPU程序的流程，提出了第一个改进方法，即分batch将矩阵送入GPU并行地计算内积。下一步同样地画出它各个步骤的运行时间占比，发现现在top-k物品的获取以及将内积结果从GPU内存复制到CPU内存这两个步骤变成了时间开销的大头。于是顺着分析结果提出了两个改进方法针对性地减小这两个步骤的时间开销。

贡献

前后提出了三个改进方法：GPU-IP、GPU-IPR、GPU-IPRO，分别为：
GPU-IP：1
GPU-IPR：1+2
GPU-IRPO：1+2+3

1. 并行计算$Q^T P$，并且提出了一种新的矩阵分割方法以充分利用GPU内存，从而加速内积的计算

   > 给定GPU内存为M，各自选取用户矩阵与物品矩阵的子集$Q_s \in Q, P_s \in P$使得$Size(Q_s^T P_s) \leq M$，论文的做法是取$Q_s = Q$，通过$Size(Q^T P_s) = M$来选取P_s的大小

2. 为每一个用户指定最佳的内积数量g_s为1024，从这1024个计算结果中返回top-k，减少了待排序的数据规模

   > 内积数量会严重影响下一步的Bitonic排序的性能。选取的依据是它应该满足每一个线程组的共享内存大小因为它会在GPU缓存层级关系中带来最小的缓存访问延迟(The size of g_s should fit in the shared memory of each threads group as it incurs minimum cache access latency in GPU cache hierarchy.)

3. 提出了一种剪枝方法来提前结束计算进程，减少了许多内积计算
   假设用户u与其第k大的物品的内积为S_k，且$||q|| \cdot ||p|| \leq S_k$，则有 $q^T p \leq ||q|| \cdot ||p|| \leq S_k$，因为目的是得到top-k物品，满足上述不等式的物品已经被排除在top-k之外，不需要送入下一次迭代进行内积计算

   > 使用这种剪枝方法后，在四个数据集的前10次迭代中，分别减少了98.88%、76.61%、88.69%以及1.57%的用户数量。