

图计算系统存储优化综述

蒋晨昱¹⁾

¹⁾(华中科技大学 计算机与科学技术学院, 武汉市 中国 430070)

摘 要 对于大型图计算系统而言, 如何存储其计算时所需要的边权节点信息, 以及如何优化存储结构, 都是值得深入思考的问题。在过去的研究中许多不同的方法被提出, 大方向如扩大存储容量, 改变存储方式, 运用核外存储, 使用分布式存储等, 而这些又能分为很多更细小更有针对性的优化方向。本篇综述围绕几篇基于图计算系统存储优化原理的论文展开, 讲述了不同图计算系统对应不同的优化方案是如何被提出、如何实施的。

关键词 图计算系统; 存储优化; 图挖掘; 图神经网络; 图划分

A survey of storage control and optimization in graph computing systems

Chenyu Jiang¹⁾

¹⁾(Department of Compute Science and Technology, Huazhong University of Technology and Science, Wuhan, China)

Abstract For a large graph computing system, how to store the edge weight node information required for its calculation and how to optimize the storage structure are all issues worthy of in-depth consideration. In the past, many different methods have been proposed, such as expanding storage capacity, changing storage mode, using off core storage, using distributed storage, etc., which can be divided into many smaller and more targeted optimization directions. This review focuses on several papers based on the storage optimization principle of graph computing systems, and describes how different graph computing systems corresponding to different optimization schemes are proposed and implemented.

Key words Graph Computing System; Storage Optimizing; Graph Mining; Graph Neural Network; Graph Partitioner

1 引言

作为一种可以表示关系的模型, 图计算实际应用中广泛用于建模和管理数据。典型的例子包括社会网络, 物理系统, 生物网络, 知识图谱等。图分析, 探索隐藏在图数据中的潜在见解, 在过去十年中引起了广泛的研究关注。它们在许多领域都发挥着重要作用, 例如, 节点分类, 链接预测, 图聚类, 推荐算法等等。

除常规图算法(如图遍历类)外, 复杂图算法(如图挖掘类、图学习类等)不断涌现。这些图算法在各个重要领域中发挥着举足轻重的作用。

在实际应用中图计算处理的数据规模往往非

常庞大。面对大量的数据, 如何设计系统内部存储方式、如何使得图算法运行速度有保证且保持存储负载均匀, 都是值得深思的课题。图计算具有数据依赖性强、访存-计算比高的特点, 提高图计算访存效率是改善系统性能的关键。尤其是随着图数据规模的扩大, 高效的存储管理优化对图计算性能的提高显得尤为重要。现实场景图计算大多具有动态变化、应用需求(例如图查询、图遍历、子图匹配、图神经网络训练和推理等)复杂多样等特征。这给图计算在基础理论、体系架构、系统软件关键技术方面提出了新的需求, 带来了新的挑战。由此呈现出的图计算体系结构和系统软件关键技术主要针对图遍历、图挖掘和图学习3类图计算应用。

本篇综述旨在通过几篇文章提出的存储思路

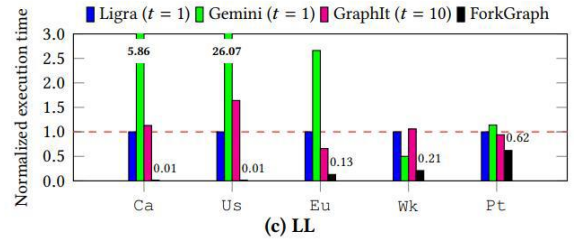
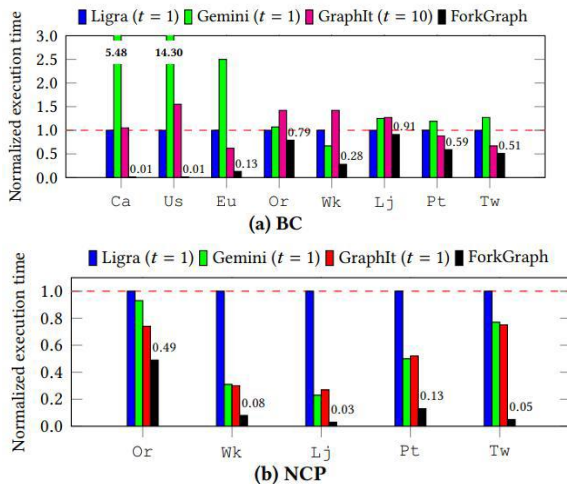
与方案来讨论存储管理优化在图计算系统中的可能性,介绍了目前先进的图计算系统硬件优化,以及面向复杂图算法(图挖掘、图学习)应用的图计算系统存储优化。

2 图计算系统硬件优化

ForkGraph[1]提出了一种缓存高效的系统,即格式图。格式图的核心设计是基于一种新的缓冲执行模型。我们将图 G 划分为LLC大小的分区,并将每个分区与存储FPP查询的操作到该分区的缓冲区关联起来。缓冲执行模型通过对来自不同查询的批处理操作来利用FPP查询之间的时间位置,并对每个分区分批执行它们。

ForkGraph包含了高效的分区内和分区间处理策略。对于分区内处理,工作效率变得至关重要,因为大多数操作都是在缓存驻留的图分区中处理的。因此,顺序实现来同时执行多个操作。通过分配一个线程来处理每个缓冲操作来利用查询间的并行性。采用顺序实现,因为它们通常比并行算法更高效。此外,ForkGraph还整合了属于同一查询的缓冲区中的操作,因此,属于同一查询的操作可以以无原子的方式进行处理。此外,这种以查询为中心的操作整合大大减少了冗余操作。

将ForkGraph与图计算系统Ligra、Gemini、GraphIT进行实验对比,分别运行NCP、BC、LL算法得到的运行时长结果如下。



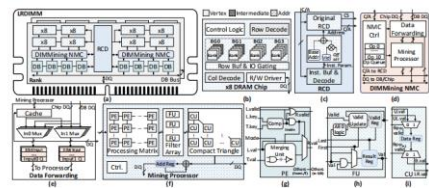
现有图划分主要分为内存划分与流划分。前者划分结果质量较高,但所需存储空间大,成本昂贵;后者划分质量较低,且一次只访问图的一条边,效率低下。HEP[2] (Hybrid Edge Partitioner) 将图划分为两块子图,一块使用内存划分,另一块使用流划分。与此同时,只有使用内存划分的子图被加载进内存中。大多数图都遵循一个规律:低度数节点多。HEP致力于减少低度数节点的复制工作。

HEP可以划分部分适合内存的图,同时产生较高的划分质量。HEP可以通过将图的边缘集分为两个子集来灵活地适应其内存开销。一个子集由NE++的一种新的高效内存算法进行划分,而另一个子集由流式方法进行划分。我们对大型真实世界图的评估表明,在许多情况下,HEP同时优于内存分区和流分区。它可以分区部分适合内存的图,同时产生高分区质量。

3 面向图挖掘类应用的图计算系统存储优化

图挖掘的问题是在输入图 $G=(V, E)$ 中找到一个给定的模式 $G'=(V', E')$ 。当前图挖掘需要面临一个挑战在于重数据传输,图挖掘需要重复访问图数据。DIMMining[3]提出了一个基于内存的近内存计算架构,可以消除计算和存储中的大容量数据转化。它基于NMC的DIMMining。DIMMining是在负载减少DIMM(LRDIMM)中使用级别级NMC处理器,不修改DRAM设备的设计,充分利用了大的内部带宽,通过避免DIMM总线争用确保了级别级计算并行性。

其大致架构如图所示。



此外, DIMMining引入了自锚和邻居分区来实

现顶点索引的预比较,提升了图挖掘中剪枝的效率,减少运行时的比较。还提出了一种灵活的BCSR(位图压缩稀疏行)格式,它工作在连续顶点上,没有内存空间开销,从数据结构的角度支持集合操作的并行性。

在实际图上的大量实验结果表明,与 fpga 和 cpu 相比, DIMMining 的速度分别提高了 222.23 倍和 139.51 倍,比最先进的图挖掘架构提高了 3.61 倍。

当前图模式挖掘 GPM 算法存在的问题:1) 为避免冗余计算,当前 GPM 工作负载采用对称性破坏,丢弃了多个数据读取,导致缓存污染和 DRAM 带宽浪费;2) 稀疏模式挖掘算法执行冗余内存读取和计算;3) 工作负载没有充分利用 DRAM 内的数据并行性。NDMiner[4]优化了 GPM 算法工作负载的性能受控和内存失速的限制,提出一种提高 GPM 工作负载性能的近数据处理(NDP)体系结构。为了减少从不同内存银行获取数据的内存数据传输,NDMiner 集成了计算单元来卸载 DRAM 缓冲芯片中的 set 操作。为了减轻对称性破坏造成的内存带宽浪费,NDMiner 在硬件中集成了负载省略单元,可以检测对称性破坏约束的可满足性,并终止不必要的负载。为了优化稀疏模式挖掘的性能,NDMiner 采用编译器优化,将减少的读取和复合计算映射到 NDP 硬件,提高了稀疏 GPM 的算法效率。

4 面向图学习类应用的图计算系统存储优化

越来越多的机器学习任务需要处理大型图形数据集,这些数据集捕获了数十亿个潜在元素之间丰富而复杂的关系。图神经网络(GNN)通过将图数据转换为低维空间,最大限度地保留结构和性质信息,构建神经网络进行训练和参考,成为解决图学习问题的有效途径。然而,提供高效的图存储和计算能力以促进 GNN 训练和开发新的 GNN 算法是具有挑战性的。

AliGraph[5]由分布式图存储、优化采样算子和运行时部分组成。AliGraph 的存储层有以下三种策略:

1) 图划分: AliGraph 平台建立在分布式环境上,整个图被划分并存储到不同的 worker 上。AliGraph 提出了四个内置的图划分算法: METIS,

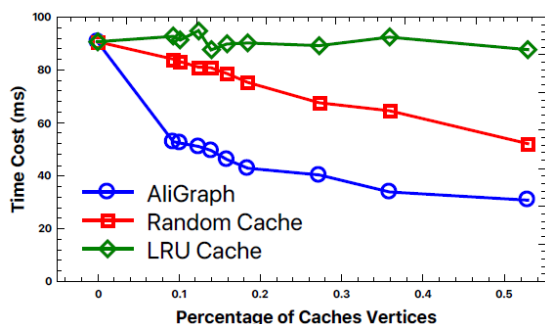
顶点切割和边缘切割分区,二维分区,流风格分区策略。这四种算法适用于不同的情况。METIS 方法专门用于处理稀疏图;顶点和边缘切割方法在密集图上表现得更好;当工人数量固定时,通常使用二维分区;流式划分方法常用于边缘更新频繁的图。用户可以根据自己的需要选择最佳的划分策略,也可以在系统中以插件的形式实现其他的图划分算法。

2) 属性的独立存储: 对于属性异构图(AHG),需要在每个 worker 中存储分区图的结构和属性。然而对于顶点和边的属性,将它们一起存储在邻接表中是不可取的。主要是因为属性占用太大的空间且不同顶点或边之间的属性有很大的重叠。因此,单独存储属性更为合理。AliGraph 构建两个索引 I_V 和 I_E

来分别存储顶点和边上的属性。 I_V 中的每个条目(I_E 响应)都是与顶点(边响应)相关联的唯一属性。并添加了两个缓存组件,分别将频繁访问的项驻留在 I_V 和 I_E 中。在每个缓存中采用最近最少使用(LRU)替换策略。缓解了单独存储属性增加检索属性的访问时间的问题。

3) 缓存重要顶点的邻居: AliGraph 提出了一种局部缓存重要顶点邻居的方法,以降低通信开销: 如果顶点 v 经常被其他顶点访问,可以在它出现的每个分区中存储 v 的外邻居。通过这样做,其他顶点通过 v 访问它们的邻居的代价可以大大降低。但是,如果 v 的邻居数量很大,存储多个 v 的邻居副本也会产生巨大的存储成本。为了更好地进行权衡,AliGraph 还定义了一个度量来评估每个顶点的重要性,这决定了一个顶点是否值得缓存。

通过实验与其它 Cache 策略比较发现,与随机策略相比, AliGraph 分别节省了大约 40%-50% 的时间,与 LRU 策略相比,分别节省了大约 50%-60% 的时间。这是因为:1) 随机选择的顶点不太可能被访问;2) LRU 策略会产生额外的成本,因为它经常替换缓存的顶点。然而,我们基于重要性的缓存顶点更有可能被其他人访问。



该系统目前部署在阿里巴巴，以支持各种业务场景，包括阿里巴巴电子商务平台的产品推荐和个性化搜索。

5 总结

本文综述了几种图计算系统的存储优化方案，展示了它们的优化思路以及实验结果。根据图计算应用领域分为了硬件和复杂图算法图挖掘、图学习三类来展示，分析了各个类的瓶颈和挑战的同时，也展示了不同类上一些优化方案。在将来的研究中，如何将图计算系统的硬件与软件优化、设施与策略优化更好地相结合，会一直是一个可用于突破图计算系统运行性能瓶颈的课题。

致谢 感谢施展老师、童薇老师与胡燚老师为我们讲授数据中心技术这门课程，也感谢课程提供的学术会议模拟机会。通过课程报告的撰写也让我对阅读论文和撰写综述报告积累了许多宝贵的经验，也巩固了许多相关的知识。

参考文献

- [1] Shengliang Lu; Sun, S.; Paul, J.; Li, Y.; Bingsheng He .Cache-Efficient Fork-Processing Patterns on Large Graphs.: SIGMOD/PODS '21:Proceedings of the 2021 International Conference on Management of Data, p 1208-21, 9 June 2021
- [2] Mayer, R.; Jacobsen, H.-A. Hybrid Edge Partitioner: Partitioning LargePower-Law Graphs under Memory Constraints, SIGMOD/PODS '21:Proceedings of the 2021 International Conference on Management of Data, p 1289-302, 9 June 2021

[3] Guohao Dai;Zhenhua Zhu;Tianyu Fu;Chiyue Wei: DIMMining: pruning-efficient and parallel graph mining on near-memory-computing, ISCA '22: Proceedings of the 49th Annual International Symposium on Computer Architecture,p 130-145,June 2022

[4] Nishil Talati;Haojie Ye;Yichen Yang;Leul Belayneh;Kuan-Yu Chen: NDMiner: accelerating graph pattern mining using near data processing, ISCA '22: Proceedings of the 49th Annual International Symposium on Computer Architecture,p 146-159,June 2022

[5] Rong Zhu;Kun Zhao;Hongxia Yang;Wei Lin;Chang Zhou: AliGraph: A Comprehensive Graph Neural Network Platform, arXiv e-prints, Feb 2020