论文总结

背景综述

本篇论文是 2018 年 USENIX ATC 会议发表的一遍论文。会议的组织者是芝加哥大学的 Haryadi Gunawi 和 Facebook 公司 Benjamin Reed。计划委员会中包含了康奈尔大学 Rachit Agarwal，卢森堡大学 Alex Biryukov 世界各国知名大学的研究学者以及微软的 Konstantinos Karanasos，Google 的 Deniz Altinbuken 等互联网公司的开发人员，我校的余华老师也参加此次会议。项目主要由 Facebook，NSF（美国国家科学基金会）资助，IBM，ORACLE 公司也给予相应的支持，主要用于研究开发新的互联网技术，推动互联网的发展。过去几次主要会议探讨存储和文件系统，云计算和边缘计算。涵盖边缘计算的新范例、新挑战、技术和商业影响，以安全、可持续的方式实现端到端（边缘-云端-边缘）的人工智能系统，如何利用人工智能技术构建大规模分布式系统，云存储，量子计算，区块链，硬件加速网络系统的趋势与未来等议题。

课题研究发展

本文主要探讨基于图的查询系统，最常用的数据库是关系数据库，将数据之间的联系用表来表示，查询时实现表的连接，将两张甚至更多的表连合并成一张表再查询，这会大大增加系统负载以及查询响应时间。因此基于图的查询系统应用而生，每个表可以看做是一个节点，这样整个数据库可以映射为一个图，用图论的搜索查询实现数据的查询，提高效率。

随着数据的不断增多，为了方便管理采用分布式的方法，数据库的不断发展也正推动着技术的不断革新。总结来说，无论采用哪种方法，目的是实现更高的吞吐量以及更低的响应时间。随着技术的更新，研究方向主要为更好的可扩展性，优化查询，数据库安全问题，云计算以及虚拟化。目前的挑战首先是安全方面问题，其次是更多的数据进入网络，如何实现更为快速精确的查找；大量的数据存储，数据中心的建设，能耗的减少，实现节能高效。在大数据环境下，结合大数据，深度学习对数据进行分析，提供更好的服务。

作者简介

Arijit Khan 是新加坡南洋理工大学的终身教授，研究方向针对大型图中出现的问题进行数据管理，重点是使用可扩展算法，机器学习技术和分布式系统，在社交和信息网络中进行用户友好，高效，近似的查询和模式挖掘。

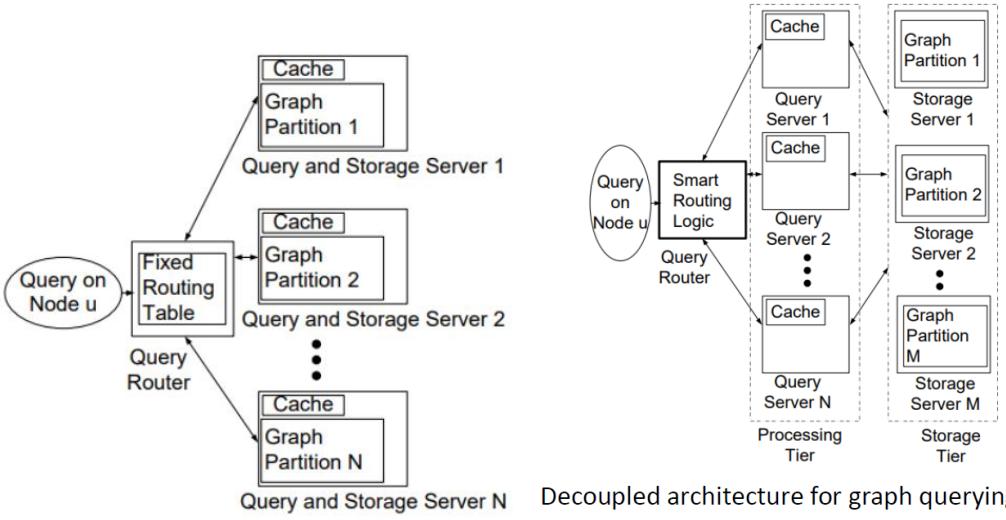
Donald Kossmann 是微软 Redmond 实验室的主任，Redmond 实验室研究计算机科学的所有核心领域。具体而言，专业领域包括算法，加密，数据库，数据分析，音频和声学，分布式系统，形式方法，图形，HCI，硬件/软件协同设计，网络（移动和数据中心网络），操作系统，编程语言，安全和软件工程。该实验室与整个公司的业务团队密切合作。此外，它与大学和科学界公开合作。作为高风险和高潜力项目的一部分，该实验室涵盖了从基础，基础研究到技术转让的整个范围。Donald Kossmann 致力于云中的数据管理，其目标是使云中的数据更便宜，更有价值，更安全。对新的数据库系统体系结构感兴趣，这些体系结构可以

提高云中数据库系统的可用性（容错性）和性能，降低成本

本文主旨

本文主要研究如何在分布式图查询系统中实现更快更有效的查询，并且提供了一种新的思路。基于图的查询系统的最终目标是实现更低的查询延迟以及更高的吞吐量，之前的方案并不灵活而且扩展性较差，本文提出了一种智能查询路由的方法，可以实现是的延迟降低，吞吐量比之前的方法高一个数量级。

随后介绍了基于图的查询系统的生活中的应用。提出解决方案，将查询处理器与图的存储实现分离，不在绑定；同时利用缓存局部性，更好的利用缓存来加快查询速率，降低延迟。首先，查询与存储的分离是逻辑层面的分离，之前的方案查询处理层与某一存储层绑定，当对某一节点的查询请求过多时，只能到一个固定的处理器处理查询请求，因为该处理器与查询的节点的存储层绑定，其他空闲的查询处理器则无法使用，加大了负载的不平衡，增加了延，因此实现处理器与存储层的逻辑分离，平衡负载，当有的处理器空闲时，直接将请求发送过去，并不需要考虑查询结果是否在该处理器查询管辖的范围之内，提高吞吐量。其次，利用缓存，当缓存中包含之前对该节点的查询结果时直接从缓存中提取，而不是再查询一次，这样降低了延迟，提高了吞吐量。



Decoupled architecture for graph querying

为新的解决方案准备条件，整个图是互联的，可以在一定步数内到达查询节点，设计逻辑层面时将存储层与处理查询层分离，存储层中设置缓存，一个存储器可以存储若干个节点的信息，处理器层也设置缓存，缓存的空间大小有限，按照 LRU 的置换算法更新缓存。设置查询路由器，将发送过来的请求发送给相应的查询处理器，同时处理器返回的查询结果也要通过路由器发送给查询方。该系统具有平衡负载，降低延迟的优点，同时缓存技术可以使得路由器更快的抉择，辅助存储器的查找。

该方案的提出从理论上层面上提升了基于图的分布式查询系统的性能，但是这也增加了其他方面的负担。例如，处理层与存储层分离后，处理层与存储层的信息交换要通过互联的网络，这会增加网络传输之间的延迟；同时当有新的存储节点插入时，要更新整个图的系统的拓扑结构，这又是一笔很大的开销，为了解决上述问题，一方面在图的划分的时候尽可能优化分区，另一方面要更好的利用缓存。

如何更好的利用缓存，最佳方案就是任何查询都能在缓存中找到结果，不用深入存储层查询，那么只能将所有的查询请求发送到同一处理器，显然是不可能的，这会加大负载的不平衡性，造成处理器资源浪费，而且排队等待时延也会增加。利用局部性原理，如果查询的

两个节点在拓扑结构上相邻很“近”，那么就将对这两个节点的查询请求发送到同一处理器。因为这两个节点相邻很近，说明有可能在同一分区，在定义节点距离远近的时候，我们采用两种方案。无论使用哪种方案，当有新的存储服务器进入重新分区时，都采用离线处理，节省时间，提高可扩展性。

1. landmark 方案：选择一小部分节点作为“地标”（landmark），计算每个节点到“地标”的距离，将每一个处理器分配给一个标记地标（pivot landmark），这些标记地标之间的距离尽可能的远，然后剩余的地标分配给处理器。定义节点到处理器的距离为节点到任何地标的距离的最小值，由于每个地标都有所属的处理器，所以地标与节点之间的距离作为节点与处理器之间的距离。这样两个相邻很近的节点有可能在同一个划分区域，之前的处理器的缓存中有可能包含其数据。同时考虑到处理器的负载，在计算距离的同时还要添加负载因子实现负载平衡。发送请求时，计算查询节点到处理器的距离，找到距离最近的处理器，由于处理器分配了一个标记地标，所以等价于查询节点到标记地标的距离。
2. Embed 欧式距离：将真个拓扑图用坐标表示，用欧式距离代替跳数，为了更加精确，可以增加坐标维数，但是并不是维数越大越好，这会导致存储空间的增加，根据实验，维数为 10 的时候效果更佳。理想情况下，我们希望每个查询都能在缓存中找到，但这是一个 NP 问题，我们无法预知哪个缓存中包含次查询请求，只能“推理”，将处理器缓存中的最近一次查询节点的坐标与此次查询节点的坐标结合一下，这时得到的坐标再去计算与处理器的欧式距离，找到距离最近的处理器发送请求。

我们进行实际测试，使用 12 台服务器，1 台用作路由，7 台用于处理器层，4 台用于存储层，4GB 缓存，10Gbps 以太网，使用在线查询，离线更新，用查询响应时间，吞吐量以及缓存命中率来衡量系统的好坏。大量实验结果表明，改进后的方案能够实现更高的吞吐量，减少延迟，但缓存并不是越大越好，当缓存不断增大时，缓存命中率会趋于一个界限，在增加缓存容量并不会显著提升命中率反而会导致浪费；同时并不是有缓存就可以减少查询相应时间，当缓存容量低于 64MB 时，缓存的内容置换算法会增加系统开销，反而会加大延迟。存储层面也可以采用新的存储方案，扩大存储容量。与现有的系统方案相比，改进后的路由算法，逻辑层面设计能够更好的处理查询请求。