数据库领域的一些学习材料

转载

iamdll 于 2014-09-01 10:29:17 发布 616 收藏

之前林仕鼎曾整理过 系统架构领域的学习资料 ，这几天Spark核心团队成员 辛湜 （Reynold Xin）公开了他整理的一份数据库学习资料列表，很有价值，Hacker News上引起了不少 讨论 。简要编译如下。大家如有补充，请评论。

基础与算法

The Five-Minute Rule Ten Years Later, and Other Computer Storage Rules of Thumb (1997): 此文与十年前的原始论文解释了一个量化公式，用来计算数据页是否应该缓存在内存中。能读到Jim Gray处理一系列相关问题（比如数据页应该多大）的方法，幸何如之。

Paxos Made Simple (2001): Paxos构成了许多分布式系统的基础。想法很简单，但理解起来却出名的难（可能是因为原始论文的写法太……）。

AlphaSort: A Cache-Sensitive Parallel External Sort (1995): 缓存友好的排序。

Patience is a Virtue: Revisiting Merge and Sort on Modern Processors (2014): 实际使用中各种排序算法及其利弊很好的综述。

关系数据库

Anatomy of a Database System (200x): Joe Hellerstein（伯克利教授，数据库专家）对关系数据库很棒的综述，涉及到各个组件。

A Relational Model of Data for Large Shared Data Banks (1970): Codd对数据独立性的探讨。尽管最近NoSQL兴起，但我相信这篇论文的一些思想在大规模并行数据系统中越来越重要了。

ARIES: A Transaction Recovery Method Supporting Fine-Granularity Locking and Partial Rollbacks Using Write-Ahead Logging (1992): 第一个实际可用的算法，支持在故障时并发事务执行而又不丢失数据。此文既有底层细节，又有高层算法的解释，因此很难读。可能还不如先去读一本数据库教材。

Efficient Locking for Concurrent Operations on B-Trees (1981)和 The R*-tree: An Efficient and Robust Access Method for Points and Rectangles (1990): B-Tree是各类数据库的核心数据结构，在随机查找时读放大因子很低。R-tree是B-Tree的扩展，支持多维数据（如地理数据）的查找。

Improved Query Performance with Variant Indexes (1997): 分析型数据库和OLTP数据库需要不同的利弊权衡方式。这反映在索引数据结构的选择上。此文讨论了许多更适合分析型数据库的索引数据结构。

On Optimistic Methods for Concurrency Control (1981): 支持并发有悲观和乐观两种方式。此文解释了乐观并发控制。……

Access Path Selection in a Relational Database Management System (1979): 查询优化的基础。此文解释了传统的成本模型方法，以及选择最佳计划的一个动态规划方法。……

Eddies: Continuously Adaptive Query Processing (2000): 此文模仿流体力学提出了一系列动态优化查询执行的技术。虽然Eddies还没有商业系统的实际应用，但很启发思路，重要性也在与日俱增。……

经典的系统设计

A History and Evaluation of System R (1981): IBM的System R和Berkeley的Ingres两个系统都证明了关系数据库是可行的。值得注意的是，30年来关系数据库的内部并没有什么太大变化。

The Google File System (2003) 和 Bigtable: A Distributed Storage System for Structured Data (2006): Google数据基础设施的两大核心组件。……虽然可能已经被Google更新的技术取代，但其中的思想将历久弥新。

Chord: A Scalable Peer-to-peer Lookup Service for Internet Applications (2001) 和 Dynamo: Amazon’s Highly Available Key-value Store (2007): Chord诞生于分布式散列表还是学术研究热点的时代。它只做一件事儿，却做到了极致：如何在完全分布式的环境（P2P）中使用一致性散列查找键的位置。Dynamo论文则解释了如何使用Chord构建分布式K-V存储。请注意Dynamo与Chord有一些设计决策上的变化，比如指取表（finger table）是O(N)的而不是O(logN)的，因为Dynamo为Amazon内部使用，对数据中心的节点有更大控制权，而Chord针对的是广域网中的P2P节点。

列式数据库

列式存储和面向列的查询引擎对于分析型负荷即OLAP至关重要，已有15年历史（最早的MonetDB论文发表于1999年），到现在几乎所有商业数据仓库都有列式引擎了。

C-Store: A Column-oriented DBMS (2005) 和 The Vertica Analytic Database: C-Store 7 Years Later (2012): C-Store是新英格兰地区多所大学（MIT、布朗、麻省等）的专家们很有影响的学术研究。Vertica是其商业化版本。

Column-Stores vs. Row-Stores: How Different Are They Really? (2012): 讨论列式存储和查询引擎的重要性。

Dremel: Interactive Analysis of Web-Scale Datasets (2010): Google令人惊叹的论文。……将列式存储应用于复杂的嵌套数据结构。论文对嵌套数据结构的支持谈得很多，对查询执行的细节涉及较少。Note that a number of open source projects are claiming they are building "Dremel". The Dremel system achieves low-latency through massive parallelism and columnar storage, so the model doesn't necessarily make sense outside Google since very few companies in the world can afford thousands of nodes for ad-hoc queries.

数据并行计算

MapReduce: Simplified Data Processing on Large Clusters (2004): MapReduce is both a programming model (borrowed from an old concept in functional programming) and a system at Google for distributed data-intensive computation. The programming model is so simple yet expressive enough to capture a wide range of programming needs. The system, coupled with the model, is fault-tolerant and scalable. It is probably fair to say that half of the academia are now working on problems heavily influenced by MapReduce.

Resilient Distributed Datasets: A Fault-Tolerant Abstraction for In-Memory Cluster Computing (2012): This is the research paper behind the Spark cluster computing project at Berkeley. Spark exposes a distributed memory abstraction called RDD, which is an immutable collection of records distributed across a cluster's memory. RDDs can be transformed using MapReduce style computations. The RDD abstraction can be orders of magnitude more efficient for workloads that exhibit strong temporal locality, e.g. query processing and iterative machine learning. Spark is an example of why it is important to separate the MapReduce programming model from its execution engine.

Shark: SQL and Rich Analytics at Scale (2013): Describes the Shark system, which is the SQL engine built on top of Spark. More importantly, the paper discusses why previous SQL on Hadoop/MapReduce query engines were slow.

Spanner (2012): Spanner is "a scalable, multi-version, globally distributed, and synchronously replicated database". The linchpin that allows all this functionality is the TrueTime API which lets Spanner order events between nodes without having them communicate. There is some speculation that the TrueTime API is very similar to a vector clock but each node has to store less data . Sadly, a paper on TrueTime is promised, but hasn't yet been released.

Dryad: Distributed Data-Parallel Programs from Sequential Building Blocks (2007): Dryad is a programming model developed at Microsoft that enables large scale dataflow programming. "The fundamental difference between the [MapReduce and Dryad] is that a Dryad application may specify an arbitrary communication DAG rather than requiring a sequence of map/distribute/sort/reduce operations".

趋势（云计算，仓库规模计算和新硬件）

A View of Cloud Computing (2010): This is THE paper on Cloud Computing. This paper discusses the economics and obstacles of cloud computing (referring to the elasticity of resources, not the consumer-facing "cloud") from a technical perspective. The obstacles presented in this paper will impact design decisions for systems running in the cloud.

The Datacenter as a Computer: An Introduction to the Design of Warehouse-Scale Machines : Google's Luiz André Barroso and Urs Hölzle explains the basics of data center hardware and software for warehouse-scale computing. There is an accompanying video . The video talks about the importance of cutting long-tail latency in massively parallel systems. The other key idea is the disaggregation of resources. Technologies such as GFS/HDFS already disaggregate disks because of high network bandwidth, but yet to see the same trend applying to DRAMs because that'd require low-latency networking.

CAP Twelve Years Later: How the "Rules" Have Changed (2012): The CAP theorem, proposed by Eric Brewer, asserts that any networked shared-data system can have only two of three desirable properties: Consistency, Availability, and Partition-Tolerance. A number of NoSQL stores reference CAP to justify their decision to sacrifice consistency. This is Eric Brewer's writeup on CAP in retrospective, explaining "'2 of 3' formulation was always misleading because it tended to oversimplify the tensions among properties."

杂项

- Reflections on Trusting Trust (1984): 1984年Ken Thompson的图灵奖演讲，描述了黑盒后门问题，指出了信任不是绝对的。

扩展阅读

许多学校都有针对研究生的数据库阅读列表

- Berkeley: http://www.eecs.berkeley.edu/GradAffairs/CS/Prelims/db.html

- Brown: http://www.cs.brown.edu/courses/cs227/papers.html
- Stanford: http://infolab.stanford.edu/db_pages/infoqual.html
- Wisconsin: http://www.cs.wisc.edu/sites/default/files/db.reading.pdf
- Joseph Hellerstein的Berkeley数据库研究生课程 阅读列表 ，比本列表更全面