

漫谈数据库的现状和未来：DTCC见闻录



孙挺Sunt

非典型性程序员 | sunt-ing.github.io

已关注

246 人赞同了该文章

这几天逛了下今年的中国数据库技术大会（DTCC），就像 @Ed Huang 黄东旭老师在会场上说的那样，感觉很多厂商年年讲的都大差不差，甚至把去年的PPT拿过来讲可能也没什么问题。**这两年数据库领域让人兴奋的东西不多，基本都是些偏engineering的东西，感觉短期内可能也未必有什么特别令人激动的东西了。**

参会时刚好遇到在代表 @DolphinDB智臾科技 摆摊的 @胡津铭 老师，和他聊了一下做数据库的方方面面，顺便聊了下我们的共同好友，某DB圈第一励志哥。

然后就是听各种各个主会场和分会场的报告。感觉好像翻来覆去就是那些名词，不由得让我想起了鲁迅《藤野先生》的开头部分。但是，我觉得细节上还是有很多不同的，而这些细节其实也对数据库的性能影响很大。所以，我还是在这篇文章中记下了我的所见所闻所想，写完数了数，快九千字了。

一点点聊吧，谈些烂大街的东西就权当是在复习，谈些“新瓶装旧酒”的东西就权当是提高鉴别力，如果能谈些“新东西”那就阿弥陀佛了。

HTAP

HTAP其实就是把TP和AP搞在一个数据库里，算是这几年的大热门了。不过各家的做法还是不太一样的。比如TiDB目前是AP和TP各跑在一套引擎上，这样AP不会影响TP的性能；而OceanBase是跑在同一套引擎上的，使得节点的资源利用率更高。

腾讯云的丁奇老师提到，**HTAP里实时性和稳定性其实不好把握。**

比如我们的TP和AP通过DTS打通，TP那边一条数据写下去后，AP那边多久能看到？现在是分钟级别。所以实时一致性很难。

稳定性也非常重要。比如你承诺TP写下数据后AP过5分钟能查出来，但是如果偶尔5分钟后查不出来呢？

单条数据写入的话AP一秒两三万，TP一秒十万，所以AP要批量写入。导入太快的话AP负载太高撑不住，导入太慢的话延迟较大实时性受到影响。所以腾讯云那边就在AP侧修改内核，暴露关键

指标和诊断信息, 包括小文件数量和请求负载等, 从而**自动调节批量大小和写入频率**, 实现写入的自动拥塞控制。DTS后续会通过旁路写入的方式, 进一步提升AP节点写入的稳定性。

丁奇老师分享了两点insight: **一个系统的复杂度就在那里, 你降低了用户的使用难度, 就增多了开发者难度; 有的需求你现在看起来好像是锦上添花, 但是过几年就是刚需了。**

超融合数据库

这个名词不知道是什么时候流行起来的, 其实就是推广了HTAP, 把TP、AP、graph、文档、全文检索、时空数据、时序数据处理等各种能力搞在一块, 还是那句话: “one size fits all”。相当于以前大家是自己在系统外“搭积木”, 现在是在一个系统内“搭积木”。

另外对于这种把不同功能拼装在一块的数据库, 我听东旭和丁奇都说过, 一个比较有意思的点在于**各个部分要有可拔插性**, 就像MySQL可以自由选择存储引擎那样。

想起了我之前就 “One size can fit all or not” 话题写过的一个回答:

[zhihu.com/question/3185...](https://www.zhihu.com/question/3185...)

NewSQL

顺便讲讲NewSQL, 或者叫Scalable SQL。OceanBase提到, **上一代的Scalable SQL牺牲了单机性能、SQL支持完整度和企业级功能**, 所以OB希望解决这些问题, 做下一代的Scalable SQL。这个是“新瓶装旧酒” 还是确有实质性不同我就知道了。本来在OceanBase定制专场听了几场的, 但还是被前面几个讲OB部署运维的talk劝退离场了, 没听到后面 @竹翁 老师的内核源码分析, 还是挺后悔的。当然, 人家DBA同学讲的还是很好的, 只是我不是DBA或运维, 也不做应用开发, 不是很care这些东西。

软硬协同

GaussDB提到了软硬协同的两点优势: 近存储并行计算, 把计算逻辑从计算节点下推到存储, 然后计算层无状态易于扩展; 有些问题单纯靠硬件很难发现, 要软硬协同。

当然, DB领域对各种modern hardwares的探索一直都有, 主要是在思考RDMA、NVM、GPU、FPGA之类的新硬件分别打破了哪些以往做数据库时的假设, 什么时候可以用, 又该怎么用?

另外, 我感觉这些新硬件真的是被用起来了。比如说NVM吧, 以前想拿这个发篇论文都不容易, 还得和Intel打好关系, 现在居然有卖NVM的商家摆起了摊。RDMA也是, 感觉现在基本是云厂商的必备了。

智能化数据库

其实就是AI for DB。智能化调参, 智能索引推荐, 慢SQL诊断等等。比如可以像腾讯DBbrain那样, 你把数据库的问题贴出来, 智能专家系统给你可能的解决办法。又比如像阿里云的那篇VLDB 2019《iBTune: individualized buffer tuning for large-scale cloud databases》一样, 自动调整云数据库buffer pool的大小。

但想把AI做到数据库内核里其实并不容易, 比如SFU今年就发了一篇VLDB, 表示learned cardinality estimation还路漫漫其修远兮。另外, 这篇文章拿了今年的VLDB EA&B best paper, 酸了。

另外, 我也听说过一些将自然语言翻译成SQL、辅助用户在一堆表中写出更好的join查询等前沿探索, 搞得花里胡哨的。

高可用部署

金融政企用户很关注高可用部署, 他们往往要求两地三中心, 异地多活, 跨地域数据分布等部署方式。

有一点还蛮有意思的, 就是蚂蚁金服和OceanBase的“三地五中心”部署, 建立起了城市级容灾的能力, 搞这个的直接原因就是几年前有个施工队一下子挖断了蚂蚁金服两根光纤的交叉口, 导致支付宝停服了两小时。

同时, 近些年各大互联网厂商在高可用服务上真的是频频翻车。去年年末, 写有《Google系统架构解密: 构建安全可靠的系统》的Google在全球范围内严重停服, 时间长达50分钟; 今年7月, 分享过《B站高可用架构》的BiliBili也崩了两三小时; 然后就是前段时间Facebook宕机了近6个小时, 给全世界科普/复习了一下BGP协议, 丢人丢得彻彻底底。

当然, 好像还没有哪个大厂没翻过车。我反正觉得互联网公司分两种: 一种是翻车了被人笑话, 另一种是翻车了也没人在乎。

高可用这东西啊, amazing。

数据库安全

数据库安全上有几点值得一提。

一个是全密态数据库。传统数据库加密是数据在传输时加密, 现在是计算时都会加密, 做到了全链路加密。

另一个是OceanBase说的。他们在内核中做数据校验, 对于数据和事务做实时校验, 对于整个副本做定期校验, 保证问题自发现。同时, 保证高并发场景性能无抖动, 并且内置灰度变更能力, 让用户放心做变更。另外, 为了防止说跑分时用弱一致性, 而用户实际用的时候使用强一致性, 所以OB不支持弱一致性, 只有强一致。听起来很强的样子。

最后是云上的数据安全保障。东旭提到: 云下的安全体系和云上是不一样的, 云下考虑权限就好, 但在云上是要考虑一整套的东西。不要重复发明轮子, 因为你自己发明的轮子都是有漏洞的。要好好利用云供应商提供的那些东西, 比如AWS提供的PrivateLink。这些都是明码标价的, 你只要把这个做到总成本里就行, 保证不亏。

老外使用数据库的方式其实和国内的不太一样。对老外来说跨数据中心真的是一种刚需: 有的数据不能出欧洲, 有的数据不能出加州, 有的数据不能出中国。为了解决这个问题还得再搭一套基础设施。那如果数据库本身就能提供这个能力, 能直接在数据库上配置, 那还是很爽的。现在国内数据库还没这样做, 但由于监管等原因这个是迟早的, TiDB未雨绸缪, 马上要发布这个功能了。

商业上的合规 (compliance) 还是很难的, 每次要认证都得掉层皮, 但是这是必须投入的成本。不然不管你这个数据库做的再怎么好, 用户也只会一票否决。

Google Cloud Spanner

这个主要还是介绍了一些Google内部和Spanner的情况, 对于我这种没怎么关注Spanner的人来说还是有点新鲜的。Google早期内部也用MySQL, 甚至给MySQL做出了不少贡献, 但MySQL跨region (为了容灾或是业务) 和横向扩展能力不行, 所以Google在2007左右就推出了Spanner。当然, 我想我们大家了解到Spanner更多地应该还是通过那篇OSDI 2012, 那个令东旭老师 “感到特别兴奋的东西”。

一开始Spanner没有SQL支持, 后来加上了。现在Google内部基本只用Spanner+Bigtable, 用得还是很香的, 大家都很开心。

Spanner的读写时延在10ms之内, 这当然比不上Redis的1ms内香, 但随着Spanner的节点数不断扩展, 这个时延始终是这么多。当然, 当资源不足时Spanner的性能也确实会有波动, 所以Spanner内部会有性能监控, 用户可以根据这个调整资源量。这一点现在还不是自动的, 未来可以做到自动。

由于Spanner用了昂贵的原子钟, 你想自己搞一套Spanner来用还是比较困难的。所以Google搞了个Spanner模拟器给大家玩玩, 然后又在2017年时推出了Spanner云服务。

Spanner云服务的客户里大概75%用MySQL, 25%用PostgreSQL。

Spanner云服务在日本还是非常火的, 日本游戏行业前十的厂商都在用。做游戏的很重视谁回档快, 由于Spanner是基于LSMT的, 所以回档会比较快。

以前国内基本没有人用Spanner云服务, 现在一些游戏厂家开始用了, 主要是在一些面向海外的业务上。不过用的人还是很少, 现场听众里很多人都听说过Spanner, 但是没有一个人用过。其实也能理解, 据我所知国内的各个云厂商其实采用的更多的还是Amazon Aurora那种架构, 像Spanner那种原子钟的做法还是太曲高和寡了。

开源和数据库

关于开源和数据库的关系, 我是觉得蛮有意思的, 后续打算专门写篇文章来聊聊。姑且举五个例子吧, 都是这次参会时刚刚听到的。

首先是华为的GaussDB。GaussDB把部分核心能力开源到openGauss社区, 希望合作伙伴基于openGauss打造发行版, 还拉了一堆用户成立openGauss社区理事会, 并且疯狂撒钱和一堆高校合作, 给我的感觉还是诚意满满的。

他们说了个让我觉得蛮有意思的点: 客户是希望从封闭生态走到开放生态, 而不是进入另一个封闭生态。想想还是挺合理的, 希望他们能做好吧。

然后是东旭, 他在讲做TiDB的insight时直言 “open source first”。基本上TiDB每一年新增的代码都会在一年内被重写, 残余的部分所剩无几, 如果没有开源的话这一点是很难想象的。

所以东旭说, 如果你现在看三年前TiDB的最佳实践文章, 那不好意思, 现在可能已经过时了。

接着是OceanBase提的两个例子。

一个是主会场上杨传辉老师提的。基于开源的产品虽然能比较快地做出产品，但很难优化到极致，因为内核优化余地有限，它不是完全自主控制的。OB在2010年从头自己做就是因为想做到最好，锻炼并沉淀工程团队能力，而不是单纯为了国产化。当然，有当前的信创战略肯定是锦上添花了。我感觉杨老师是自信满满。

另一个是OB定制场上纪军祥老师提的。OB曾在2014年开源过，然后又闭源了，这其实给大家留下了很不好的印象。所以OB不可能再去闭源，因为如果再次闭源了，整个OB、蚂蚁甚至阿里在开源社区的信誉就要扫地了，以后他们再做开源有谁信呢？同时，OB表示，蚂蚁内部将开源分为了四个等级，OB是唯一一个最高级的。

我是觉得，少看一个人说了什么，多看他做了什么。目前看来OceanBase的开源还是很真诚的，周围小伙伴都说好。

最后是一家主要面向金融领域的数据库公司。他们告诉我，由于金融领域的特殊属性，他们没有开源，将来也没有开源计划。

刚进互联网领域的时候，我常常感觉，一个数据库如果不开源，那它是基本没有人用的，因为大家选型的时候可能根本就不会用你，甚至都不知道你。但是，在DB-Engines Ranking上，Oracle依旧排名第一，世界范围内最流行。另外，纪军祥老师说，今年DB-Engines Ranking上的开源数据库的数量第一次超过了闭源数据库的。纪老师想以此说明开源力量的日益兴旺，但我反倒凭直觉觉得三五年前就应该超过了才对吧。看来还是我太年轻啊，见的世面少。

另外，我联想到前段时间的那篇ICSE 2021，它对国内BAT的开源状况做了一些分析：《An Empirical Study of the Landscape of Open Source Projects in Baidu, Alibaba, and Tencent》

云和数据库

东旭老师表示，虽然这两年数据库领域没有特别令人兴奋的东西，但是有一个很重要的改变，那就是云。因为现在大家都假设我的软件是直接跑在硬件上的，但是这个假设改变了，可能5年后我们就不需要关注底下是什么硬件，是什么CPU，什么硬盘了。将来我们的孩子关注的可能是AWS有S3、Lambda，它们有哪些API之类的。

这两年东旭老师一直致力于cloud，而非TiDB的内核，他觉得内核也就那样了，反倒是整个数据库行业都低估了云的影响。他引用Gartner的话说：企业架构师应该拒绝那些非cloud-first的产品，不是cloud-first是没有机会的。

华为也给出预测说，到2025年，中国大中型金融政企会占到云市场3/4的份额。虽然公有云价格低，但客户出于监管和安全考虑，会倾向于部署私有云。

那么，当我们在聊cloud-native时，我们在聊什么？其实只有一件事情：cost。

你在云上的生意其实是规模化生意。现在很多数据库供应商遇到大客户就想着上点一体机，这种卖盒子的做法是不scalable的。很多数据库供应商把高科技干成了施工队，碰到个大客户就巴不得一下子塞20个人来响应需求，但是这个做法不scalable。一个DB公司只有cloud-native，才能做成千亿美金市值。

东旭老师自黑一下TiDB。TiDB诞生的时候，云还没那么火，所以TiDB用的是share-nothing架构，存算切的不干净，对资源利用率比较低，被大家吐槽。当年也想过计算存储是不是要分开来，但当时那只是一个基于工程的朴素思想。

然后东旭老师聊了很多云上的基础设施。

- GP3: 它是block storage, 很神奇的是, 对于它, 你有多少钱, 就能买多少IOPS, 这在云下是不可想象的。
- Spot instance: Spot instance虽然不稳定, 但很便宜, 我们可以好好利用它, 用它来做一些即使跑挂了也无所谓的处理计算。
- S3: S3是令人发指的便宜, 存1TB数据1个月不到20美元。如果我们自己做一个S3, 基本上是亏钱的。东旭老师说的这一点还是很可信的, Snowflake早期也是自己做云储存的, 但是发现不管怎么搞就是做不过AWS S3, 然后就放弃挣扎了。
- Serverless: 这个是用来做计算的, 也非常便宜, 可以好好利用起来, “薅一薅云厂商的羊毛”。我感觉Serverless现在确实是一个热点, 不管在学术界还是工业界都非常火。
- 存算分离: 只有存算分离吗? 不, 能分离的都可以分离, 都可以自由扩展, 包括存储、缓存、网络、CPU等。因为一个应用对于各种资源的要求是不同的, 以前没有云的时候其实造成了很多的浪费, 现在借助云做这样的拆分会让你有很高的自由度, 成本上能大大节省。
- Kubernetes: 要搞云原生的数据库得重写Operator的很多逻辑。东旭老师提的这一点和我听云数据库实践专场时听到的差不多。
- Pulumi: 虽然整个大会场只有三个人举手表示听说过Pulumi, 但东旭老师对于它还是很看重的。目前每家云厂商提供的SDK都大差不差, 但又不完全一样。可以利用Pulumi写编排脚本, 凡是能自动做的, 就不用让人来做, 要搞基础设施代码化 (IaC)。另外在云上, 数据库厂商一定要提供跨云的平滑迁移能力。

东旭老师反复强调, **在云上真正值钱的东西是CPU, 而不是I/O; I/O不是bound, CPU才是。**

可以说, 整个会场所有talk里东旭老师的PPT是最简陋的, 白纸黑字没有任何装饰, 但我觉得对我的启发也是最多的。

当然, 我觉得数据库厂商做云也有做云的问题, 是否开源、怎么开源、license是什么、和哪些云供应商合作、怎么合作.....这些问题一定要尽早想好。相关的事件其实还是有不少的, 比如ElasticSearch、MongoDB以及一些国内的数据库公司和云供应商的纷争, 其中一些内幕消息甚至不足为外人道也。

面向数据科学的数据库

这个是百度的马如悦老师讲的, 科普了一下面向数据科学的数据库。我不知道为什么talk中没有提DuckDB, 据我所知DuckDB是这个领域第一个吃螃蟹的。不过讲的还是很好的, 虽然东西对我来说并不算新, 但还是让当时又饿又累又困的我集中起了注意力。他说这个领域是前沿方向, 说Databricks在做, Snowflake可能也要做。据我所知, DuckDB恰好几天前成立了商业公司。

和我们互联网人可能不是很熟悉金融场景一样, 我们可能同样也不熟悉数据分析师的需求。对于大规模数据分析, 我们天然的会想到Presto、Spark, 如果要做机器学习, 那就Mahout、Spark MLlib之类的。

可是数据科学家们对于这些东西兴趣寡然, 因为它们在公司里太难申请、太难调通、太难操作了。怎么有这么麻烦的下载、调试和权限认证? 怎么那么多配置? JVM内存、堆外内存又是什么? 我就

打算花十分钟分析个表怎么用得着这么麻烦? 像Snowflake之所以要做云服务, 原因之一在于就方便使用, 免去手工部署运维环节。

数据科学家喜欢的是在Jupyter Notebook里用Pandas之类的Python库、R库。但问题在于, Pandas在优化上做的比较差: 比如它假设你的数据全在内存里, 如果你要分析100GB的数据就得有100GB的内存; 又比如它没有太多的查询优化, 一些filter下推优化都没有做。而这些问题吧, 数据库领域几十年前就做烂了。

同时, 现在硬件性能已经很高了, 一台电脑的性能可能抵得上10年前100个节点的集群。数据科学家的数据量一般在500GB内, 其实单机就可以搞定。再想想ClickHouse, 它的缺点其实非常明显, 但它一下子就爆火了, 这并不是因为它可以无缝替代MySQL, 而是因为它把单机性能优化到了极致, 而各种分布式数据库中单节点的性能往往比较低。讲这一点时, 马老师联动起了OceanBase的杨传辉老师, 老友互掐, 整个talk一下子就有意思起来了。

可以说, **我们很多时候需要的不是分布式数据库, 而是优化得更好的单机数据库。**

另外, 还有几个点值得思考。

对于数据集成, 大数据从业者一般是使用ETL这种集中式的集成, 这种做法发源于传统的数仓, 而数据科学普遍接受的是联邦查询。

还有, 数据分析领域的逻辑如果写成SQL可能高达几千行, 很难阅读和修改, 解决办法就是搞点延迟计算和SQL DAG。

另外, 断点续跑、远程数据本地缓存、避免重复查询等都可以去做。

其他

其他一些零零碎碎的东西也记在这吧。

一个是主会场主持人玩的一个文字游戏, 说数据库领域要做到“人、工、智、能”: “人”表示由人来做出挑战突破, 牵引需求; “工”表示工具, 即利用工具降低工作量和成本; “智”表示智慧、自治和自动, 即降低人为干预; “能”表示赋能业务发展, 惠及整个行业。

然后是东旭结合做TiDB的感受说了几点insight:

protocol > implementation: 用户根本不关心底下的数据库是什么, 是MySQL还是MariaDB, 最重要的是MySQL的那个协议, 所以TiDB选择MySQL协议就赢了, 因为客户最多;

usability matters!: 和很多人想的或者很多友商做的不一样, TiDB做一个决定时很少从技术出发, 而是关注用户体验, 希望用户用的爽。**因为很多用户其实不太关注你TPC-C跑多少分, 而是关心用的爽不爽, 用户体验才是最重要的。**

总结

感觉这些年数据库领域包含的东西是越来越多了, MySQL是数据库, MongoDB是数据库, HBase是数据库, Redis是数据库, 连ElasticSearch也是数据库, 什么时候连HDFS、Pandas甚至Tensorflow也算是数据库呢?

从根本上说, 数据库是查询引擎和储存系统的结合, 但现在好像单独的查询引擎或储存系统都可以算是数据库了, 它们的边界在一点点模糊。

从这次的DTCC会议中也可以看出点迹象：以前DTCC里没有多少讲大数据系统的人，现在有一堆人在讲，大数据和数据库的界限也是越来越模糊了。

数据库短期内的发展方向也很明确了。

可以利用新硬件，探索怎么在数据库里利用NVM、RDMA、FPGA等；

可以利用云服务，让数据库变成云原生的，这也算是一种特殊的利用新硬件吧；

可以探索新场景，思考流处理、向量检索、图处理等特定需求；

可以增强易用性，把TP、AP、图处理等引擎结合，或者简化使用数据库的一些流程，降低它的使用门槛；

可以增强数据库智能化，让数据库根据数据的变化自动调整，而非由DBA手工调，这也部分属于增强数据库的易用性吧。

我总有一种感觉，现在确实是数据库发展的黄金年代，特别是国产数据库发展的黄金年代。

人才建设方面，国内数据库领域第一次涌现出了这么多这么杰出的人才，而且是同时覆盖了老中青三代，我感觉对此平凯星辰公司 @PingCAP “难辞其咎”；

政策支持方面，由于前段时间的各种国际形势，国家推出了信创战略，这让很多政企金融部门向国产数据库敞开了大门；

资源投入方面，由于Snowflake和MongoDB等数据库公司的股价一飞冲天，无数热钱疯狂涌入数据库领域，大家都想从中分一杯羹。

天时地利人和，国产数据库的风口终于来了，就看能不能把握住了。

编辑于 10-22

[数据库](#) [数据库系统](#) [MySQL](#)

已赞同 246  25 条评论  分享  喜欢  收藏  申请转载 ...

文章被以下专栏收录



Singularities 的数据库随想
做梦都想读的数据库专栏