不同角度下的分布式数据库应用特点

该文以TiDB和MySQL作为对比进行阐述。主要以不同角度分析使用MySQL（单机）与TiDB（分布式）有哪些不同作为切入点，如何更好的应用分布式数据库？在思维上需要有哪些转变。

一、关系型分布式数据库 VS 关系性单机数据库

1、关系型单机数据库

优势：

1、**成熟稳定：**经过40余年的发展，几乎应用到了所有的行业，已经被打磨的非常成熟稳定，生态完善；

2、**行业适配性强：**适配不同行业的各种需求；

3、**生态完善：**有大量的ISV应用开发商和技术开发者，技术生态，产业生态和人才生态都很完善；

劣势：

1、**高成本：**核心场景需要依赖高端硬件，软件售价高；

2、**无法横向扩展：**只能纵向提升硬件资源（CPU/内存/磁盘/网络，或者小型机/大型机）获取更高性能；

挑战：

1、单机硬件要求高，部署成本高，硬件维护成本高昂，实现横向扩展能力成本代价过高；

2、应对高峰期高并发，大流量场景能力较弱，当突破单机能力上限，剧烈的资源争抢等会导致整体性能极具下降。

2、基于关系型数据库的分库分表方案

优势：

1、**线性扩展能力：**通过分库分表实现数据库水平扩展；

2、**系统迁移成本低：**数据库引擎延用，无需考虑语法兼容等问题；

劣势：

1、**跨库分库分表事务：**原有数据库引擎基本无分布式能力，只能依赖中间件，但中间件很难做到RPO=0，异常情况下无法完全保证分布式事务的ACID能力；

2、**MVCC全局一致性：**多个服务器时钟不一致时，无法保证多个库之间数据版本号的全局一致性；

3、**副本均衡：**扩容或缩容时，无法在线调整数据分布规则，需要暂停业务手工重新分布数据，业务和运维成本很高；

4、**跨库复杂SQL：**跨库的复杂SQL查询（比如多表做分片键无关的关联查询只能在中间件完成，而中间件不具备分布式并行计算能力），这将限制应用对SQL的使用，产生业务侵入性；

5、**分片键：**需要人为选择分片键，业务SQL也需加分片键查询，对业务产生侵入性；

6、**数据强一致性：**部分分库分表采用cdc方式实现副本复制，中间件无法保障数据强一致性，只能保障最终一致性。

3、原生分布式关系型数据库

优势：

1、**数据与服务天然高可用：**基于多数派协议（raft/paxso，实现副本一致性）的工业级实现，个别节点故障时可保证数据零丢失（RPO=0）和服务快速恢复（RTO<10秒）

2、**线性扩容：**可随着业务的发展，峰期的变化，灵活扩缩容

3、**低成本：**基于普通x86或arm服务器保证高可用，无需高端的小型机和存储

4、**全局一致性：**支持分布式事务，确保全局一致性，支持分布式复杂查询

5、**灵活的部署方式：**支持同城双中心，三中心，多中心部署模式

6、**对业务透明：**开发者可像单机数据库一样使用分布式数据库，业务迁移改造成本低

7、**线性提升能力：**吞吐量通过横向扩展可以线性提升

劣势：

1、**产品成熟度：**成熟度不如单机数据库

2、**运维困难：**dba需具备分布式系统维护知识

3、**延迟高：**相较于单机数据库，网络等消耗较大

二、理解分布式数据库

分布式系统常见的挑战有哪些?

1. 故障与部分失效
2. 不可靠的网络
3. 不可靠的时钟
4. 数据复制强一致性
5. 分布式事务
6. 数据均衡分布

7. 一致性共识
8. 元数据瓶颈及管理

三、开发者角度

常常会遇到一个问题，虽然TiDB高度兼容MySQL协议，但在TiDB的开发上有哪些注意事项?

1、热点问题

TiDB以region为单位进行数据计算存储复制，默认96M，基于rang进行数据切分，这种方式在高并发的批量写入便会遇到热点问题，同样在高并发的小范围条件查询时也易出现热点问题。

解释起来就是，我的并发写入或读取的数据都在某一个或几个region中，那么对于分布式而言，一直使用的都是某一台机器或者两台机器的资源，无法充分发挥出分布式系统的优点。

下面介绍一下常见的解决思路：

对于写热点：

1、对于业务上依赖auto_increment属性的id字段，尽量替换为auto_random，目的是写入时可将连续性分散，从而发挥出分布式系统的特点。

2、提前为热点表规划多个region，发挥出分布式系统的特性；

3、将region进行切割，默认96M，切为更小的region，减小热点持续时间；

4、通过where条件，打散数据，高并发写入；

对于读热点：

读热点而言，往往容易出现在判断条件小范围或点查，高并发场景下，如几千并发查询的都是某台机器上的region数据，这样就容易造成单台机器瓶颈。下面的一些解决措施需要根据业务特性来使用，并不是万能之法。

1、切割region，将所查询的region数据分散到多台机器；

2、将部分查询分散到副本上进行，发挥出分布式系统特点，利用多副本特性，将负载分散开；

3、对于高频查询，低频更新的小表而言，将表缓存到内存中；

4、对于跑批程序，适当增大分段数，从而减小并发；

综上所述，站在开发者角度，应尽量让程序的数据分散存储，并发分散，充分利用多台机器的资源。

2、充分发挥出分布式高吞吐能力

分布式数据库对高并发友好，但需要一定前提，比如单纯的point get，几万并发，分布式数据库性能必然比单机差。所谓的高并发友好，是指高并发查询或更新涉及多张表，大范围的数据，充分利用起各台机器资源的前提下。

在对几个亿甚至几十亿数据量的表做无条件查询时，分布式数据库可轻松胜过单机数据库，这就是发挥出了分布式系统多台机器资源的能力。

同上面描述相同，应尽量让应用程序利用到分布式系统的优点（多台机器同时计算）。

3、索引

与单机数据库相同，所有SQL都应有合适的执行计划，选择合适的索引。只是分布式数据库的缺点是，严重依赖网络，尤其对于TiDB这种计算存储分离的架构，点那个索引不合适时，将给网络IO造成灾难性的结果，比如网络堵塞，从而拖慢整个库的性能。

所有SQL都应该走合适的索引，并且为了减少回表，尽量使用聚簇索引！

4、大事务

同样的，在大事务的SQL时，对网络IO的消耗也是很大的，最好方式将大事务进行拆分多个事务，高并发的执行。

尽量避免大事务，做好事务的拆分，这对数据cdc同步也很重要。其实这在单机数据库中的cdc数据同步时，也应尽量避免大事务。

四、DBA角度

1、部署

分布式数据库多采用自动化运维工具（如ansible）进行部署管理，应对自动化运维工具有一定来了解，这在以后的部署，巡检，升级等都很重要。

2、备份

除了传统的数据备份外，还应考虑分布式数据库各类管理工具文件的备份，以及分布式数据库元数据的备份等

3、负载均衡

相较于单机数据库的数据倾斜而言，分布式数据库应关注各类资源（如cpu/内存/磁盘io/连接数等）的使用是否均衡，一般情况下，都会有不均衡情况出现，需要根据制定一个界限，有针对不均衡情况的处理措施。其实也可将该点理解为热点问题。

五、基础架构角度

1、服务器的时钟

对分布式数据库而言，时钟是很重要的点，多为MVCC版本号所依赖，如时钟出现错误，可能会导致数据不一致等问题出现，应对服务器的时钟准确性充分关注。

2、网络波动

分布式数据库需要多台机器组成，这对网络的稳定性，延迟，吞吐都有较高要求，应尽量保障网络的稳定。

3、服务器故障

因为分布式数据库，天生的高可用和故障恢复能力，因此当集群出现单台机器故障时，无需过于紧张。

4、合理的资源使用

对于像TiDB这种存储计算分离架构的分布式数据库，可考虑分配不同的资源。

tidb计算实例分配高频赫兹的CPU处理器，普通SSD即可。

tikv存储实例分配高性能（如nvme）磁盘。

pd调度实例在保障网络的情况下，16vcore32G普通盘500G虚拟机即可。

当然具体资源分配，还需根据业务特点以及要求制定！