今天和大家分享一篇SIGMOD2019

作者认为区块链和分布式数据库系统有类似之处。the Case of Hyperledger Fabric，他们研究了fabric平台，发现可以用分布式系统中的一些比较经典的技术去改善Fabric区块链的性能。
作者在论文中比较了区块链和数据库系统的相同和不同的设计理念。

在分布式系统中，一个隐含的假设是每个节点都是可信的。但是在区块链系统中是可以存在恶意节点的。
那我们是否可以认为，区块链系统是下一代 分布式系统呢？
作者不认同这个观点，因为现有的区块链系统有很多部分，比如共识、激励机制、智能合约等，
其中交易处理模型是比较原始、落后的。

交易处理模型关心，交易的生命周期，交易从哪里来，到哪里去，要经过哪几个阶段的处理。
现有的比如以太坊比特币这样的公链，他们的交易处理模型，被称作排序执行模型。

我们来完整看一下一个完整的交易生命周期是什么样子的。
可以看到交易是客户端发起的，然后由排序服务把一笔一笔交易打包成块，每个块里可以有上百笔交易。
最后由一些节点去执行交易。然后把区块的数据，以链的形式记录下来。

但是这样的模型下，交易是无法并发执行的，必须顺序一笔一笔执行交易。
这个问题可以用数据库已有的机制去解决。

具体来说就是在排序之前加入一个模拟执行的阶段。
然后把执行阶段，换成 验证提交的一种机制

那么这样一种：模拟执行、排序验证提交的交易生命周期，是文章研究的联盟链fabric采用的。
在模拟执行阶段，可以做到交易的并行化。是一种乐观的并发控制。
但是在最后一个阶段有可能产生序列化冲突。
需要相应的机制去处理这种冲突。

我们具体讲一下序列化冲突是什么。
ABC有一个初始值 5 3 7 。
在模拟执行的阶段，黄色交易是write
写A的值为10。蓝色交易读A的值为5，写B的值为8。红色交易读A的值为5，写c的值为12

经过排序阶段，一个区块中这些交易顺序是正好是123。

因为后两笔交易的读集是一个旧的值，所以后两笔交易被标记无效。这种情况下的成功率只有三分之一。其余交易是失败的。

如果把交易执行的顺序重新排一下。成功率100% 。
思路是一个非常简单的思路。

那么作者改进的系统fabric++就是采用了这种交易重排技术。

首先用图论的方法，构建冲突交易图。规则就是：如果交易I写的key，是交易J读入的key。那么Ti指向Tj。

如果有六笔交易，他们之间的关系是这样的。那么首先计算出强连接的子图。
注意到：子图里的交易是冲突的关系。必然存在一些需要被丢弃的交易。
然后在这些子图之间，计算无环冲突的图。最后得出交易的顺序。
在这个例子里，顺序就是T5，T1，T3，T4。T2和T0会被丢弃掉。这些交易可以正常执行。

接着我们来看原生的Fabric是如何处理序列化冲突的。它采用的是一种基于锁的并发控制。

我们重新强调一下交易的概念。
每一笔交易可以更新键值对。

比如一个X，可以索引到它的值是70。它是由第二个块的T4交易更新的。
类似的Y，它的值是80，是第一个块的T3交易更新的。current state代表了当前键值对的最新状态。
描述这个新旧状态的，是一个版本号。版本号，是哪个区块的哪笔交易更新的。

我们观察一笔T6交易，在模拟执行阶段，它读到XY的版本是T4和T3。
但同时节点验证了第三个区块的T5交易，这笔交易将X和Y的版本都更新成T5。

当T6交易被打包进第四个区块，在节点处进行验证阶段。
它读取的版本号和当前世界状态不一致，因此被丢弃。

Fabric++采用了一种更加精确的并发控制。
首先在验证阶段，确定节点当前最后一个区块的高度是多少。在这个例子里是T4所在的区块。
然后模拟执行的交易读到X的版本为T4。

恰巧这个时候节点来了一个新的区块，区块中的交易被验证后，将X的版本号更新为T5 。

如果这个时候，模拟执行的交易进行读Y操作，读到的版本为T5,比T4要大。
直接就在模拟执行阶段丢弃这个交易。
因为这笔交易就算被打包进区块并进入验证阶段，也会因为读集和世界状态不匹配而被丢弃。

这样就可以解决模拟执行和验证的在节点处的并发操作。

在Fabric++里面，目标是尽量提前无效交易的丢弃操作。
原本交易的丢弃发生在最后一个阶段。
排序服务，做交易重排的时候可以丢弃无效交易，提升交易的成功率。
在simulation模拟执行阶段，可以使用前面提到的多版本并发控制，丢弃无效交易。
提升整体的交易性能。

改写了Fabric的源代码，并在分布式网络上进行性能测试。

资产转移的一个工作负载。有6种交易类型，5个更新，1个读。一些可变参数，写交易百分比，Zipf数据分布。
另一个工作负载，参数更加具体。

执行这些工作负载，跟原生的fabric性能去做对比。采用重排、早期丢弃技术的fabric++性能有明显的提升。
（跳过一页）

把这几个技术分开来看他们对性能提升的效果。
采用重排技术的性能、采用早期丢弃的性能。结合两个技术的性能。

采用交易重排技术能够将交易成功率提升12倍。减少百分之五十的延迟。
工作主要贡献，关注到了一个典型的交易重排问题，提高交易成功率。
并且对交易生命周期去做了分析、对应地使用一个基于版本的并发控制技术，提高交易性能。