FISCO BCOS的速度与激情：性能优化方案最全解密

原创 石翔 FISCO BCOS开源社区 2019-04-19

FISCO BCOS

系列专题 | 区块链的“慢”和优化之路（2）

FISCO BCOS的性能优化

作者：石翔

石翔

FISCO BCOS核心开发者

再小的性能问题都是问题

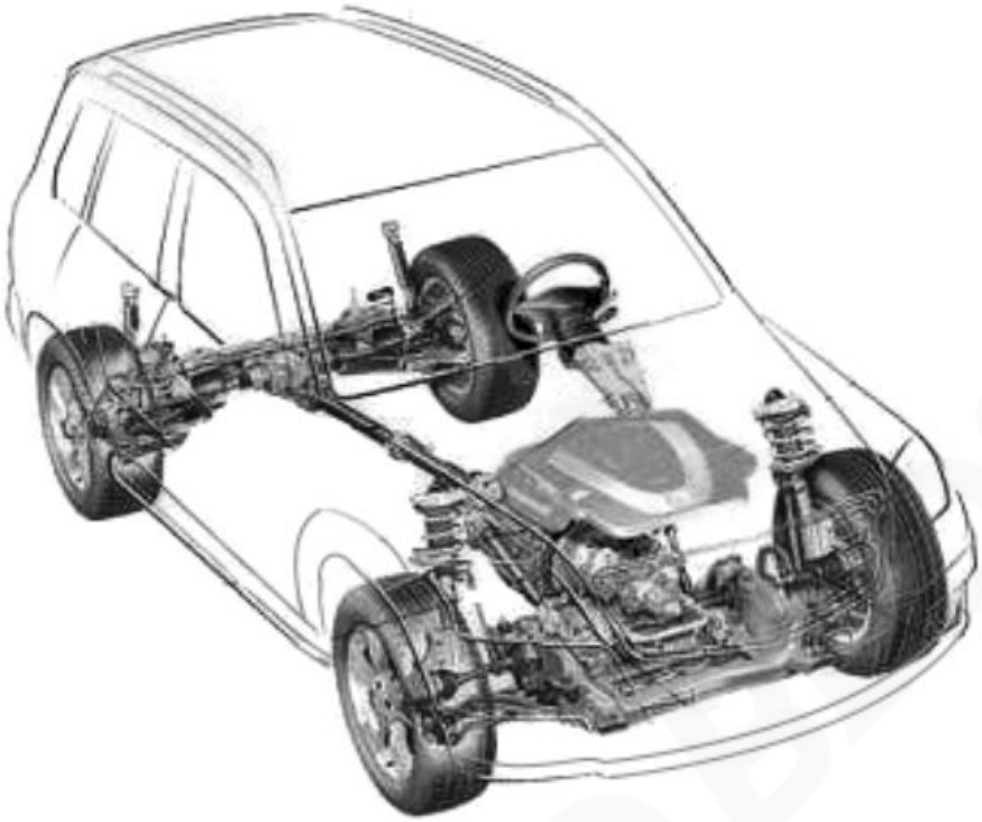
AUTHOR | 作者

上篇文章说到，区块链的速度困境是“贵”在信任，“慢”得其所，说到底，根因还是在其“用计算换信任”的设计思路上。业内普遍赞誉区块链是信任的机器，为了实现信任，区块链不得不做很多复杂而繁琐的操作，同步、验签、执行、共识等，都是区块链中必不可少的环节。

这就像行车时的“交规”，时刻在告诉我们开发者，为了安全，请按规定速度行驶！然而，社区里大家还是有着共同的心声：真的太慢了！

> 道路千万条，安全第一条
> 行车不规范，亲人两行泪

那么，能不能对这台信任的机器来一次装备升级，让它既安全又快速呢？经过团队深入的探索和实践，我们打通了多条迈向极速时代的路子。回看整个过程，恰如打造了一台性能卓越的汽车：



高功率发动机☞**基于DAG的交易并行执行引擎**
燃料输送装置☞**分布式存储**
前排与后座☞**共识与同步的流程优化**
传动装置☞**全方位并行处理**
氢燃料☞**预编译合约**
监控仪表☞**全面的性能分析工具**
专属的方向盘☞**可并行合约开发框架**

高功率发动机：基于DAG的交易并行执行引擎

——极尽所能让交易并行执行

传统的交易执行引擎，采用串行的方式执行交易，交易只能被一条条依次执行。一个区块中无论有多少交易，都需要一条条依次执行完。这就好比一个低功率的发动机，纵使给它装上巨无霸油箱，仍然无法输出强大的动力。

1个气缸不够，改成4个气缸，8个气缸，行不行?

FISCO BCOS实现了一种交易并行执行引擎（PTE），能够让一个区块内的多个交易同时被执行。若机器有4个核，最大限度能支持4笔交易同时执行，如果有8个核，则能支持8笔交易同时执行。

当然，在“交规”管控下，并行执行的正确性需要得到保证，也就是说，并行执行的结果和串行执行的结果需要一致。为了保证并行执行的一致性，FISCO BCOS的交易并行执行引擎（PTE）引入了DAG（有向无环图）这个数据结构。

执行引擎在执行区块中的交易之前，会根据交易相互间的互斥关系，自动构建交易间的依赖关系。这种依赖关系是一种DAG，在引擎执行时，会根据DAG让可并行的交易并行执行。这样一来，交易执行的一致性得以保证，交易执行的吞吐量也得到数量级的提升。

燃料输送装置：分布式存储

——为引擎提供足够的燃料

传统的区块链存储模式，是一棵参天的MPT树。区块链上所有的数据，都汇聚到这棵树上来。对数据的每一次写或读，都是一次从树枝到树根（或者从树根到树枝）的漫长旅行。随着链上的数据越来越多，树也越来越高，树枝到树根的路程会变得越来越长。

更麻烦的是，虽然树枝有很多个，但是树根只有一个。对海量链上数据的写或读，就像千军万马抢过独木桥一样悲壮，惨烈程度可想而知。所以传统的区块链，选择了一个个来，一个数据一个数据地读，一条交易一条交易地执行。形象地说，就是用一根输油管为引擎输送燃料。

这样肯定不行！我们需要多个输油管为引擎输送燃料！

这一次，FISCO BCOS不是粗暴地为引擎接上多个输油管（MPT树），因为用输油管输油（用MPT存数据）实在是太慢了。我们干脆抛弃输油管，直接把引擎泡进油箱里！这样的比喻也许欠妥当，但理解了FISCO BCOS的执行引擎和存储设计，相信你会和我有一样的感慨。

我们抛弃MPT树，采用“表”的方式组织数据。执行引擎读写数据，无需再对MPT树进行树根到树枝的遍历，直接在“表”上读写。这样一来，每一条数据的读写，都不依赖于一个全局的操作，可以分开独立进行。这就为交易并行执行引擎（PTE）提供了并发数据读写的基础，类似于泡在油箱里的发动机，汽油直接流入气缸，谁也不共用谁的输油管。

分布式存储详细解析请点击：分布式存储架构设计

前排与后座：共识与同步的流程优化

——不搞平均主义，先富带动后富

在区块链节点中，同步模块和共识模块，是一对形影不离的双胞胎，有时相互帮助，有时也为争夺资源大打出手。在以往的设计中，同步模块和共识模块并没有优先级的区分。好比坐车，谁坐前排，谁坐后排，没个规定，导致这对双胞胎经常在争夺先后顺序上浪费大量的时间。

一切从实际出发，先富带动后富！

共识模块负责主导整个区块链出块的节奏，应让共识模块先行。而同步模块，理应扮演好配合的角色，辅佐共识模块更快出块。基于此思想，FISCO BCOS对共识与同步的流程进行了优化：

第一，将同步模块中交易验签的操作，从P2P的回调线程中剥离出来，让共识模块能够更加顺畅地收到共识消息，以便更快进行共识。

第二，对交易验签进行去重，并对交易的二进制进行缓存。一笔交易只进行一次验签和解码，为共识模块中区块的执行腾出更多的CPU资源。

第三，优化同步流程，在交易同步之前，尽可能地让同步模块跑在共识模块之前，从而使得同步模块优先把交易写入交易池中，优先进行解码和验签，让共识模块拿到交易时，免去解码和验签的过程，更快进入区块打包阶段。

总而言之，言而总之。一切的目的，都是为共识的流程服务，让其更快更顺畅地打包、执行、共识、出块。

传动装置：全方位并行处理

——让功率有效地输出

若不搭配合适的传动装置，再高功率的引擎也无法将功率有效输出。签名验证、编解码、数据落盘，是区块链中除开交易以外，其他耗时占用较高的部分。在以往的设计中，签名验证、编解码、数据落盘，都是串行执行的。就算交易被并行执行，这台信任机器的性能，也受制于这三个环节的性能。

这三个环节的性能问题一日不绝，性能永无抬头之日！那就给高功率发动机配上一个高性能传动装置，释放出它的威力来。

FISCO BCOS引入了并行容器，让数据的读写天然支持并发访问。在此基础上，对于交易的验签，直接让交易的验签并行执行，交易与交易间的验签流程互不影响；对于编解码，改造了RLP的编码格式，使原来只能按顺序读写的RLP格式支持并行的编解码；对于区块落盘，对状态的改变进行并行编码。

不仅如此，FISCO BCOS在可并行之处都进行了并行处理，让系统CPU资源得到最大化利用。交易不仅在进入合约引擎时能并行执行，在诸如签名验证、编解码、数据落盘等环节中也都是被并行处理的。强大的发动机，配合上高性能的传动装置，果然效果显著啊！

氢燃料：预编译合约

——高效率的轻量级合约框架

众所周知，区块链上跑的是智能合约，智能合约用solidity语言编写。solidity合约部署到链上，烧掉Gas，得到结果。但是，有没有想过换一种燃料，一种成本更低却又让车跑得更快的燃料?

且看FISCO BCOS自研的“氢燃料”——预编译合约！

FISCO BCOS为机构提供了一种高性能、定制化、轻量级合约框架。机构可按照自身业务需求，将自己实现的预编译合约内置于FISCO BCOS节点中。

预编译合约用C++编写，其性能高于solidity引擎，而且启动速度更快、指令更精简、内存使用更少。正如“氢燃料”一般，成本更低，热值更高，让汽车跑得更快！当然，提取“氢燃料”需要下一点小功夫，预编译合约的实现相对复杂，门槛比较高。

了解预编译合约请点击：预编译合约架构设计

监控仪表：多维度性能分析工具

——给人全局在握的踏实感

FISCO BCOS在开发过程中使用了大量的性能分析工具，就像汽车上安装了诸多指数清晰的监控仪表。

我们采用了主流的性能分析工具，如perf，systemtap，对程序的热点、锁、内存等进行了分析，还根据区块链的程序流程特点，开发了定制化的性能分析工具，以便在共识，区块验证、存储模块和流程中更好地评估数据。

工具能够对程序中各个阶段的时间占比、时间变化进行分析。有了可靠的量化工具，开发者在做每一处优化时，都能做到心中有数。

专属的方向盘：可并行合约开发框架

——给开发者流畅的操作体验

一切准备就绪，上车！

坐上驾驶位，你手中掌控的，将是FISCO BCOS为你提供的专属方向盘——可并行合约开发框架！如何合理操作这台机器，全靠这个方向盘。

“手握”并行合约开发框架，在开发并行合约时，合约开发者无需关心具体的底层逻辑，而是将更多注意力集中在自己的合约逻辑上。当合约部署成功后，并行合约会被底层代码自动识别，自动并行地执行！

现在，终于开上了车。没过瘾？没关系，接下来的几篇，请接好，是真正的硬核干货！我们将在下一篇文章中，系统介绍**FISCO BCOS中基于DAG模型的并行交易执行器（PTE）**，敬请期待~

FISCO BCOS

FISCO BCOS的代码完全开源且免费

下载地址↓↓↓

https://github.com/fisco-bcos

长按“二维码”关注

点**阅读原文**，获取更多FISCO BCOS开发教程

阅读原文