

# Infura 系统结构简要分析

时间 / 版本	主要功能	备注
2022.01.20	Infura 系统结构	

编写：程华峰

## 1 Infura 测试数据

有 2 个简单的数据：

A) 官网公开 Infura 公开测试数据：

- a. 平均接口调用时间：236ms ~ 255ms
- b. Endpoints and services: 99.41% ~ 99.99 % uptime (in 90days)

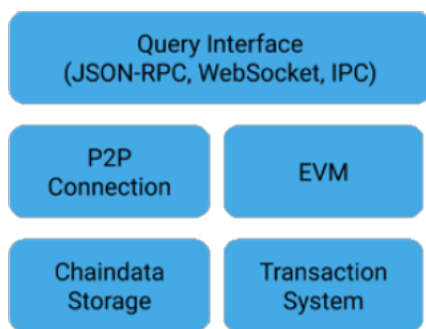
B) 使用 QuickNode 测出 Eth Mainnet 接口时间对比 (30s 测试总时间)：

接口	QuickNode 响应时间	Infura 测试响应时间
eth_getBalance	174.0ms	372.2ms
eth_getCode	169.7ms	353.3ms
eth_blockNumber	168.7ms	384.5ms

## 2 Single Node Architecture

单节点服务模式，就目前看是 Harmony.one 出现节点频繁断开服务的主要原因。单节点是由单个节点提供服务，所有的开发者使用相同的 endpoint。

单节点提供服务的模型如下：



Node 暴露 endpoint 给外部调用者 (Dapps)，由于节点本身吞吐率的性能，单节点在遇到频繁请求接口会出现服务不稳定 / 断开服务的情况。

还有一种办法是通过在子系统中设置单独的故障域，可以单独地对子系统进行优化：

- \* 使用反向代理实现接口调用的 load balancing
- \* 使用防火墙可以实现防范 ddos 的攻击
- \* 增加节点数量来提升接口的高可用和提升接口请求的吞吐率

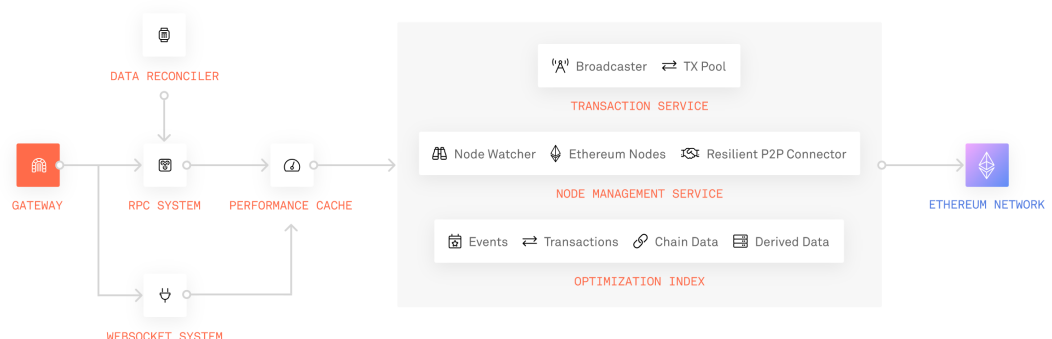
## 3 Infura's Cloud Architecture Model

### 3.1 Infura 结构



Infura 的 Service layer (Ferryman) 提供服务：API Service、Node Service、DB Service。

另有结构图：



\* 经过 Load Balancing (Gateway) 之后的请求分为 RPC 、Websocket 请求

\* 请求先经过 Performance Cache，Infura 使用 Redis 集群作为缓存

\* Transaction Service 有两个组件：Tx Pool、Broadcaster。此处的 Tx Pool 通常认为是 Eth 节点内部的 txpool，不确定 Infura 是否有将此部分做成服务的形式，Broadcaster 将需要广播的交易传递进 Txpool，即可以广播到链上传播

\* Node Management Service 有三个组件：Node Watcher、Eth Nodes、P2P Connector。（待补充）

\* **Optimization Index** 有 4 个部分：Events、Transactions、Chain Data、Derived Data。Tx（合约交易触发 emit event）在节点内执行时，会生成 Events；Chain Data 是链数据；Derived Data 是由链数据做的辅助数据。

## 3.2 Infura 做的优化

\* 针对 **Events**，Eth 的内部接口为 **eth\_getLogs**，其会在内部使用 Bloom Filter 来预先判断数据是否在 Eth 的 World State 中生成，但 Infura 发现在处理大量请求时，Bloom Filter 在进行检索时会增加数据加载的时间并减慢请求响应的速度，所以 Infura 构建了 Log Indexer。Log Indexer 会把新区块中生成的 Event 数据单独保存做索引，减少了 **eth\_getLogs** 接口的响应时间。同时限制每次请求 log 的时间不超过 10s、每次返回的 logs 不超过 10k 条。

\* 针对 Eth 的 **re-org**，Eth 会生成叔块（分叉），所以每次同步到一个块时，不确定后续的状态会不会因为分叉而导致状态改变（链状态数据改变，导致的 DB 中的索引数据的改变）。Infura 做了 **reorg Tracker**，来监听 Eth 是否发生 Re-org。

\* 对于请求的处理，Infura 做了优化。

- a) 请求 Redis 缓存数据；
- b) 如果 redis 找不到，再去 Specialized Indexers 请求；
- c) 再找不到，直接找 Node 节点直接请求。

\* **Project ID** 和 **Project Secret** 的引入，增加接口调用安全性。

\* **Vipnode** 的服务，将服务节点与内部节点连接，有助于减少区块、交易的传播时间和提升带宽利用率。

\* **Node Monitor**：监控 Node 运行状态，一般监控的数据由区块高度、P2P Peers 的数量，如有必要可以重新启动节点。

以上。