## 区块链的数据存储

### 基于键值对的数据存储模式

其中一个方向为基于键值对的存储模式。该模式下存储的任一数据记录包含一个可以确定该记录的主键，并将其余部分视为该主键所对应的数值进行存储。该模式数据结构简单，可以在处理大规模数据时获得很好的性能。考虑到部分共识协议需要大量的网络开销，在数据规模极大的情况下，直接将所有数据存储于区块链中将产生巨大的存储开销和网络通讯代价。因此部分工作旨在以较小的代价提高数据存储规模，包括分布式存储、链下存储等技术。

### 基于关系型数据的存储模式

另一个方向是采用关系型数据的存储模式。该模式将数据以关系模型的方式建模并存储。在该模式下，可以保证数据满足诸多限定条件。例如数据库的一致性、原子性等。

## 区块链事务处理

### 确保原子性的处理技术

[63]通过排序来确保事务的原子性。[74]通过锁机制建图来确保事务的原子性。

### 确保隔离性的处理技术

Hyperledger Fabric采用先模拟后排序的思路，整个流程包含了模拟阶段、排序阶段、验证阶段、执行阶段四部分。并在验证阶段检查冲突。但此方法导致大量事务被终止，并发度降低。[64]提出了利用冲突图的方法产生执行顺序的算法，通过查找其中最大的无环图，产生相应的事务执行顺序。[65]和[66]基于Hyperledger Fabric研究了事务的并行执行问题，通过建立关系依赖图，确保数据一致性。以上方法是从排序阶段和执行阶段解决冲突。[67]和[68]则在验证阶段检测事务的冲突。

### 确保一致性的处理技术

文中提出了在传统的先排序后执行并发控制和先执行后排序的并发控制方式两种场景，确保事务执行的顺序一致。

## 区块链查询处理

### 一般查询处理

[82]构建额外的图结构支持溯源，[83]利用智能合约确定数据关系。[81]提供类似于SQL查询语句来对区块链数据进行查询。

### 可信查询处理

[84] 假定用户节点并不一定存储整个区块链的全部数据，而是只存储所有区块中的哈希值。在每个区块中添加一个额外的命名为AttDigest 的字段实现可信查询。[85]研究在以太坊中如何减少开销。

## 区块链可扩展性

### 基于分片的区块链系统

本文着重介绍了[42]、[43]、[87]三种技术。[42] 利用分片提高区块链可扩展性，[43]在上述基础上不需要所有节点都存储区块链的全部数据。[87]解决了[42]在访问数据时的死锁现象。[88]则主要解决分片环境下数据的一致性问题。[89]则解决分片环境下算力分散问题。

### 区块链的链下事务处理

该方法只适用于节点之间可信的场景。[102, 103]分别基于比特币和以太坊,提出了支持比特币的Lighting网络和支持以太币的Raiden链下支付网络。但是链下支付网络存在再平衡的问题。用户需先存于一定的金额于支付通道中。并且该金额只能用于该支付通道中的两个用户之间的支付。若用户同时想于另一节点交易，但拥有的金额不够，则需通过已经建立的通道进行交易。[106]通过对交易通道建模成线性规划问题，设计了基于环形结构的再平衡算法。