链滴

图解比原链 Tensority 算法：如何让 POW 做到人工智能友好

作者：bytom

原文链接：https://ld246.com/article/1562209594016

来源网站：链滴

共识算法说起

区块链系统首先是分布式系统，而一致性是分布式系统的基础问题，要保证系统满足不同程度的一致，则就要用到共识算法。

现在主流的算法有POW、POS、DPOS等等，比特币采用的POW共识算法运行9年之久，已被证明稳可靠，然而因为巨大的硬件和能源消耗而饱受诟病，特别是专用矿机，在被淘汰之后就变成了废铁。

POS和DPOS为了避免资源的浪费，直接采取抛弃计算的方式，通过持有证明和选举来进行共识，牺了一定准入性和去中心化。而比原链从另一个角度来切入和解决POW资源浪费的问题。

比原链共识算法Tensority设计思路

首先我们基于以下思路来设计共识算法：

- 计算是一种权力，不能因为能源消耗而抛弃计算的方式，为了维持系统的稳定能源消耗是必要的，且POW已经被证明稳定可靠，同时准入门槛低。（CPU、GPU、ASIC矿机皆可进入，同时也不需要币或者选举）
- 矿机功能单一是原罪，只能进行哈希运算造成矿机的极大浪费，如果能够将矿机功能多样化将更有于发展。
- 人工智能技术的迅猛发展，AI智能加速市场需求量增大。

为此我们设计了AI友好型的共识算法Tensority。矩阵乘运算与卷积运算是人工智能常用的两种算法相比后者，前者的应用范围更广。

为了使得比原链共识算法对人工智能友好，同时兼容所有主流的AI加速设备，比原链在算法选型上采了矩阵乘运算。算法确定之后需要选取参与运算的数据类型，选取的标准有二：

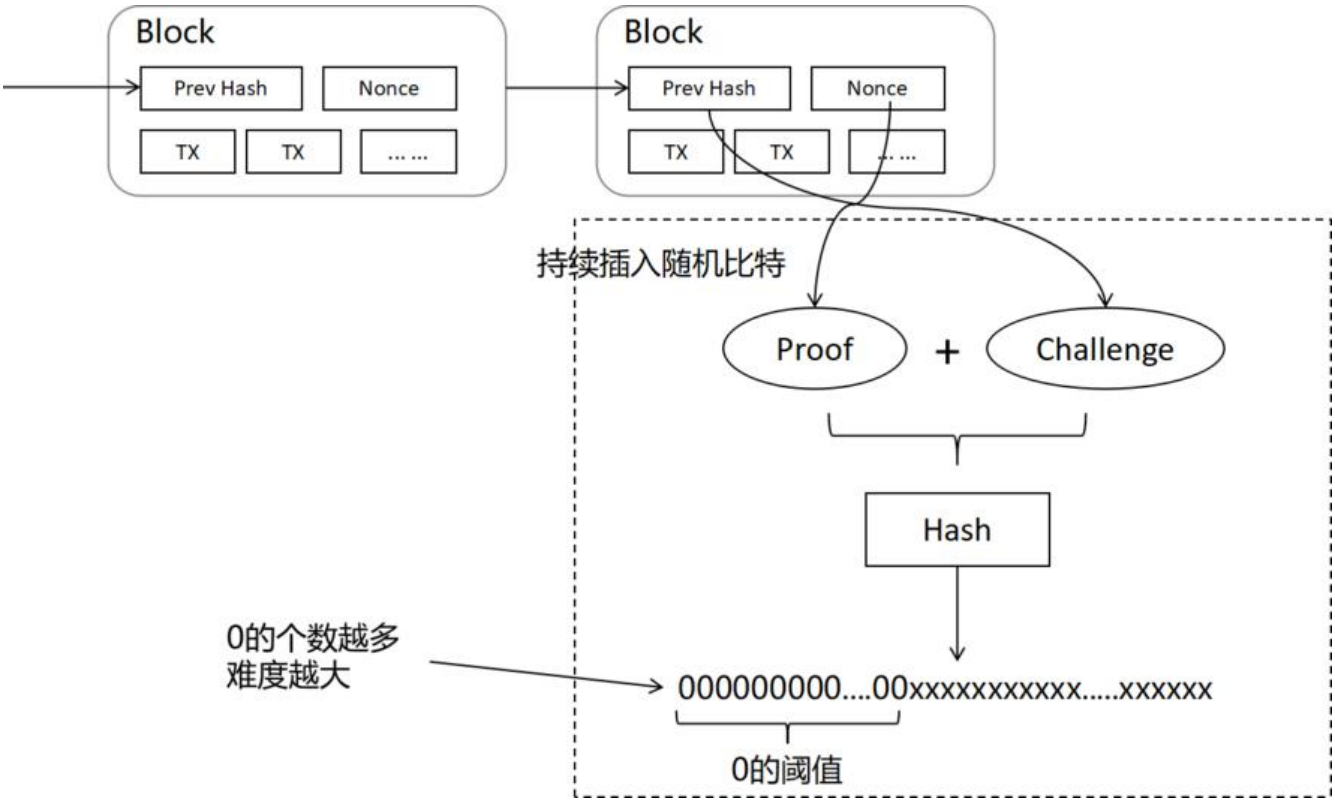
第一，选取的数据类型需目前所有主流AI加速设备都能支持。

第二，神经网络推理的主力数据类型均要支持。

综合来看，int8数据类型满足条件。

比特币POW共识算法回顾

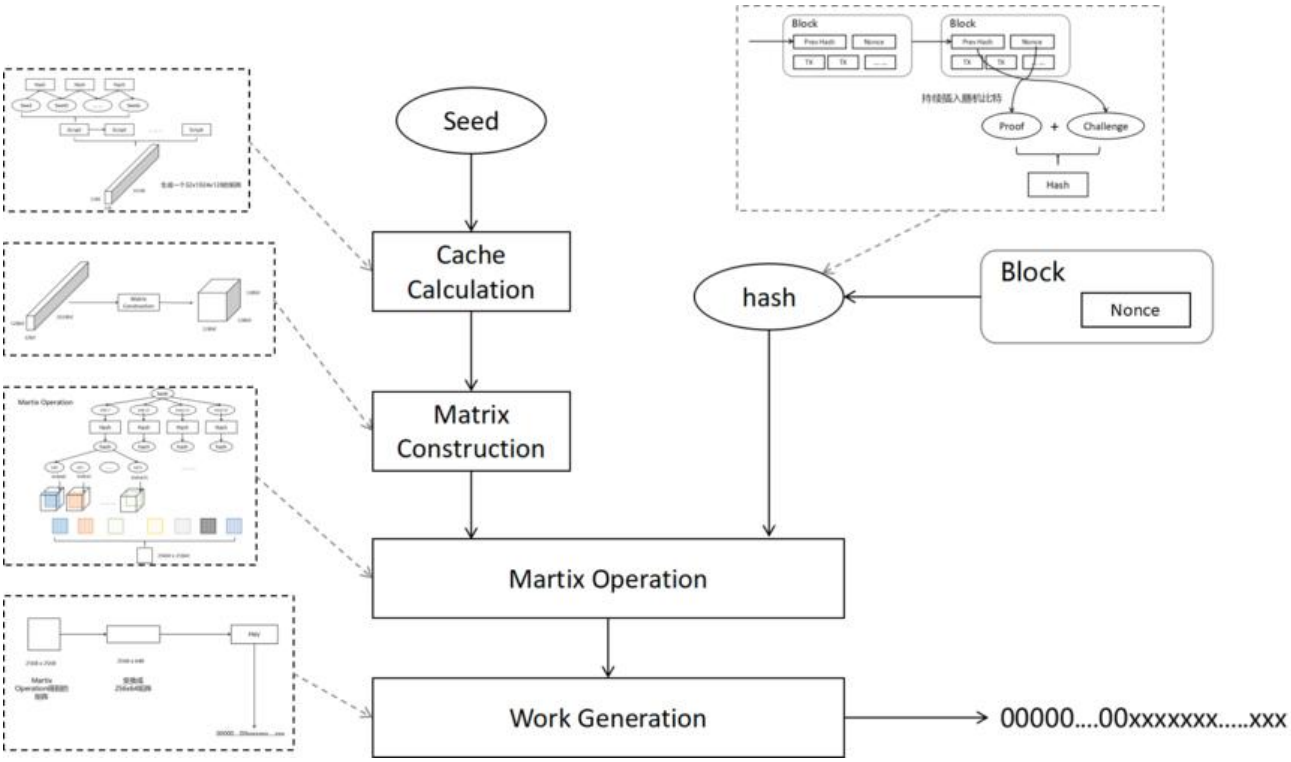
在说到比原链的POW共识算法Tensority之前，我们回顾一下比特币的POW共识算法：



我们知道比特币的POW共识算法是通过不停的迭代计算区块头的哈希值，不断修改参数，直到哈希匹配的过程。

比原链POW共识算法总览

那么让我们看一下比原链的共识算法总体过程：



整个Tensority算法过程中，区块头的哈希的选取和难度值的比较仍然作为头尾衔接的步骤，但是中穿插了很多涉及到矩阵的运算过程，而这些运算在AI计算中比较常见，所以支持比原挖矿的矿机就有

原文链接：图解比原链 Tensority 算法：如何让 POW 做到人工智能友好

力提高AI加速服务。

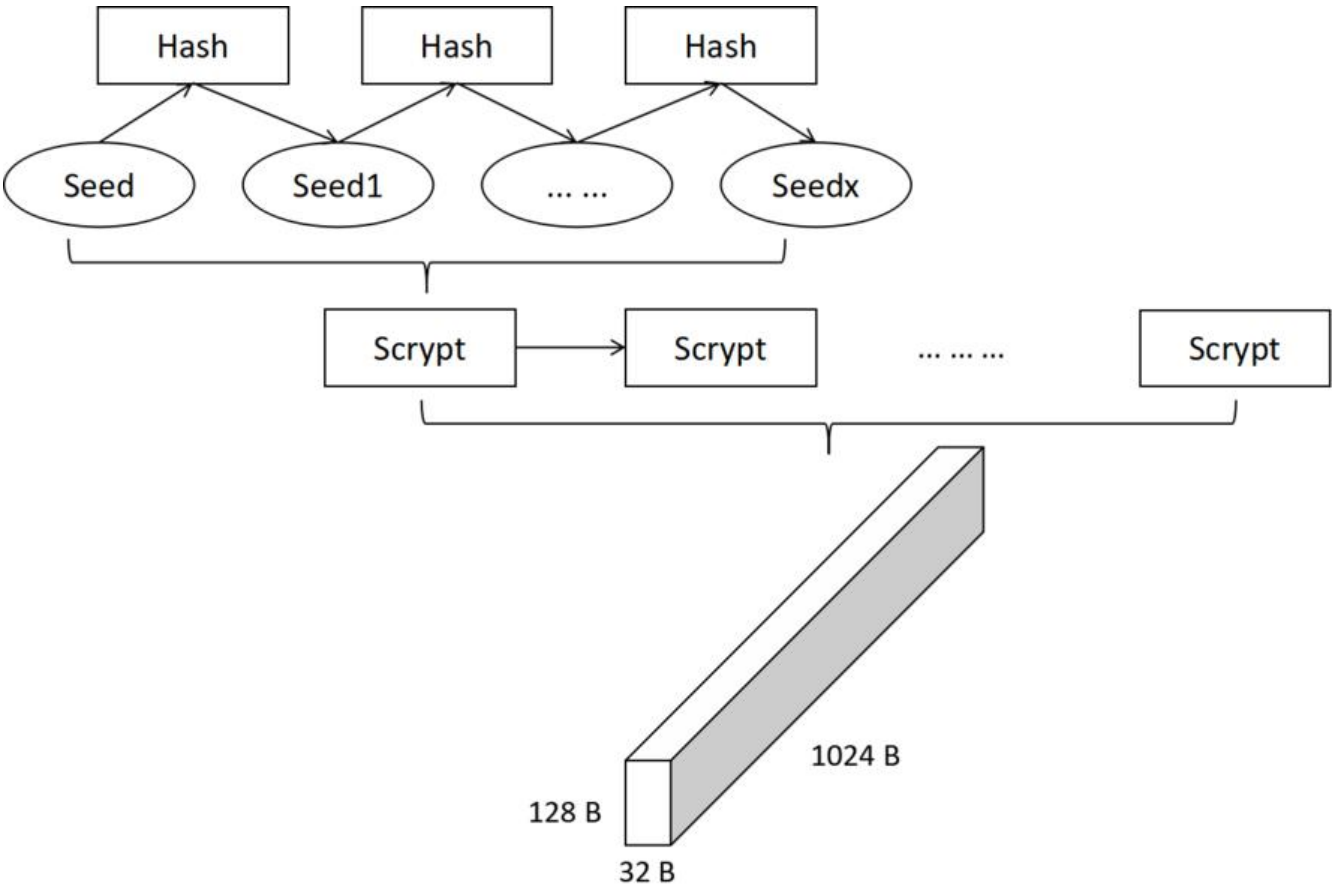
下面我们来进一步细化每个过程：

种子生成

我们在总览图中可以看到Tensority有两个输入，一个是和比特币相同的哈希头，另外一个就是种子seed，那么种子是怎么来的？ 我们可以看到论文中对于种子的描述：种子是一个由一段时间内的区块历决定的32位的字节数组。种子来自于每256块的第一个块的区块头，一般来说每256个块会更换一次seed，在256块以内都会使用相同的seed。

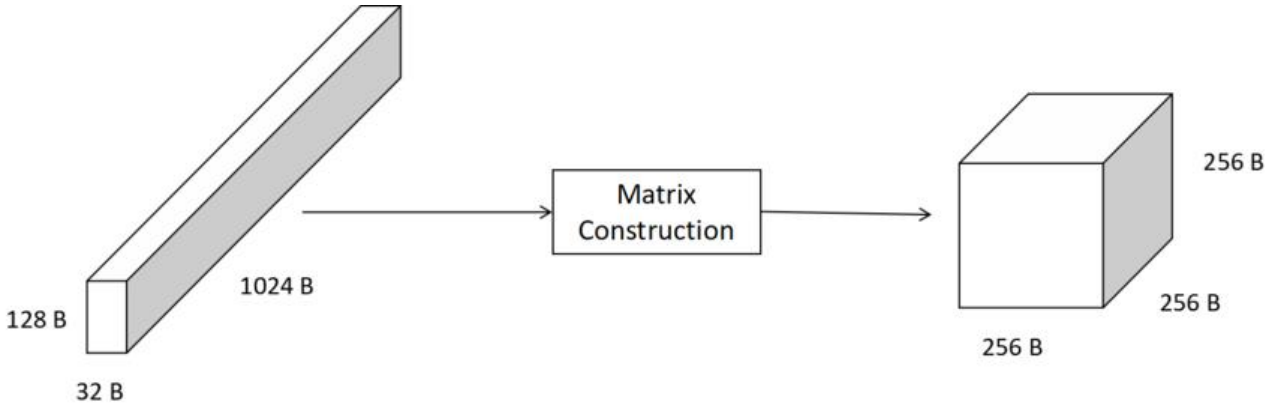
Cache Calculation

这个步骤主要使用种子通过一定的变换获得一个矩阵。我们首先通过一定次数的哈希将种子进行扩展以满足Scrypt的输入要求，然后使用Scrypt函数生成一个32x1024x128的矩阵。值得注意的是，我们用的Scrypt算法就是使用在莱特币中的算法。



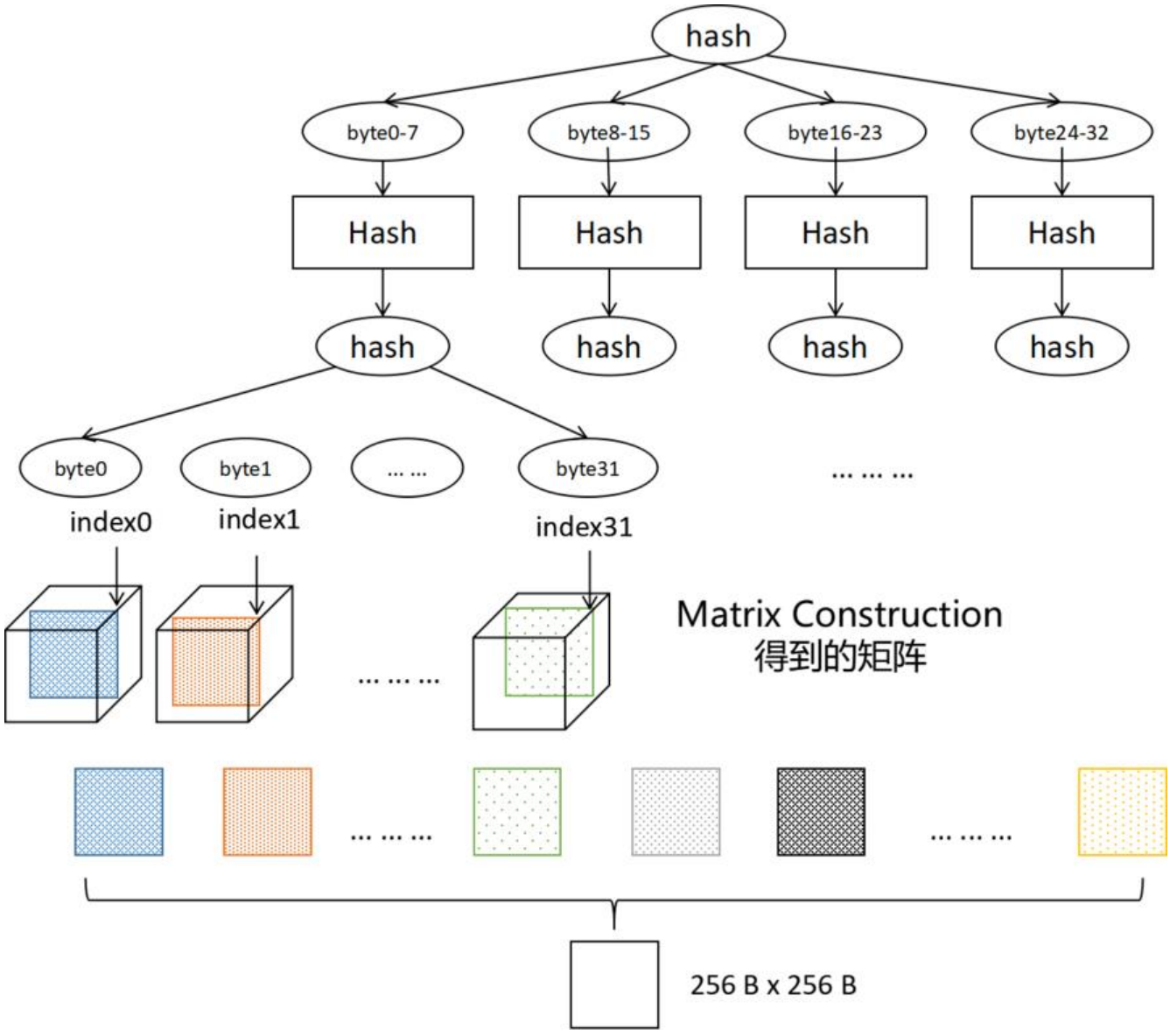
Matrix Construction

该步骤会将上一步产生的矩阵变成一个更为规整的矩阵从而适合后续的处理，具体过程将会比较复杂请参考论文。



Martix Operation

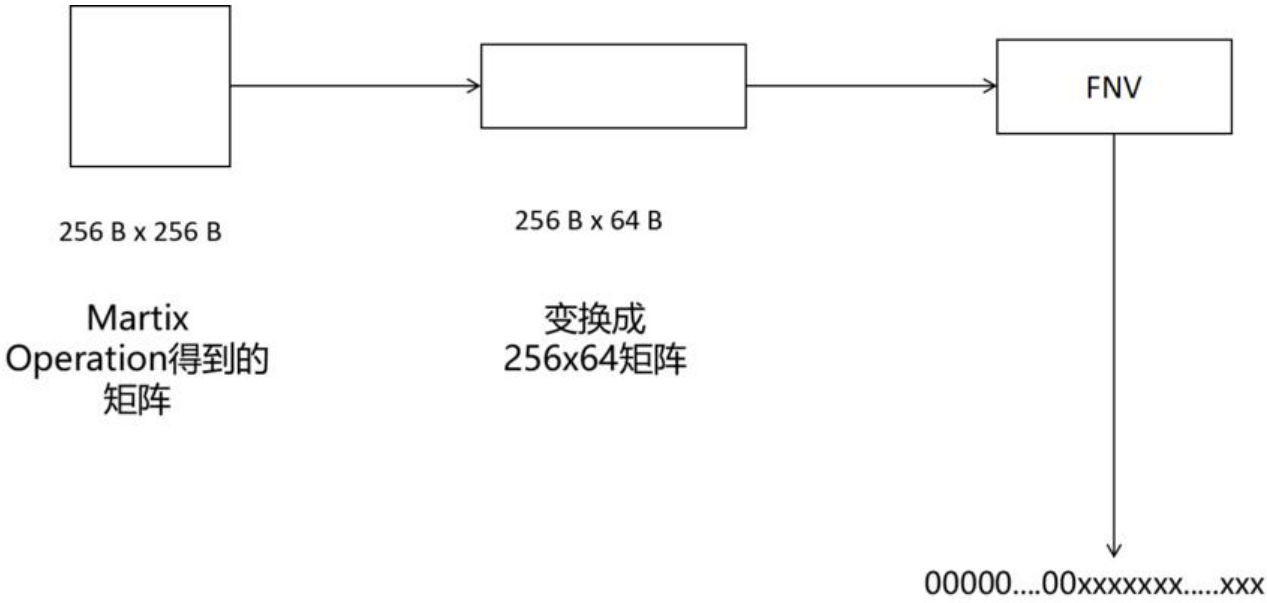
该步骤比较复杂，也是最有意思的步骤。采集的区块头哈希分割为四份，每份做一次哈希生成一个新哈希值，新的哈希值的每一个比特作为Matrix Construction生成矩阵切片的索引值，从而切片获得个矩形。经过上述步骤后将获得128个矩形，对这些矩形进行矩阵相乘最后得到一个矩阵。



Work Generation

原文链接：图解比原链 Tensority 算法：如何让 POW 做到人工智能友好

这个步骤是输入上一步生成的矩形变成一个32位的哈希值，从而进行最后的难度比较。首先将256x25的矩形变形位256x64的矩形，然后通过FNV函数转换为一个32位哈希值。



我们将得到的哈希值和难度值做比较，看是否满足条件，这一轮的共识算法就结束了。

原文链接：图解比原链 Tensority 算法：如何让 POW 做到人工智能友好