SIMPLIFIED COST SCALING ANALYSIS FOR NON-MINING BITCOIN NODE SITES.

WATERHOUSE

1. Introduction

We present here a simplified costing analysis for the network of non-mining, fully-validating, relaying node sites of the bitcoin network. Quantifying costs for a single site is difficult because they vary widely by geographic location and site configuration. However, a total network cost approach is used here that maybe useful to extract some scaling conclusions. A breakdown of component costs for a site and a look at the bottleneck of bandwidth costs scalings is also instructive.

2. Analysis

The total number of bitcoin node sites is given by N; note we use sites to refer to the autonomous installation of power, communication and maintenance requirements, regardless of how many node instances may be running on a site. Whilst additional software nodes on the same site may be desirable for the site user, they don't increase the network's physical redundancy. For some total cost to operate the network, C_{tot}, under the given assumptions, we can then arrive at a simplistic average cost per node, $\widetilde{C_n}$ where

$$C_{tot} = N \times \widetilde{C_n}. \tag{1}$$

Simply put, the total cost of an asynchronous, decentralised network scales linearly with the number of sites; the costs of highly distributed physical redundancy increase proportionately. Equally, any increases (or savings) in the average cost per site affect the total network cost proportionately.

It's worth considering that, to within an order of magnitude, the total cost the network as an entity is willing to bear, out of altruism or ancillary participation benefits beyond mining rewards, maybe mostly constant and the number of sites adjusts to accomodate that based on average site cost inputs. In that case, scaling transaction volume up will be met with a proportionate decrease in number of sites. Or similarly, a per node cost saving due to technical improvement, or an increased ancillary benefit, will see an increase in the number of sites.

2.1. Component costs.

The cost for a site operating a node is simplified to be the sum of the component costs

$$C_n = C_c + C_s + C_r + C_b + C_o \tag{2}$$

where

- C_c - cpu computational costs (tx and block validation)
- C_s - storage costs (blockchain)
- C_r - RAM costs (mempool, utxo)
- C_b - bandwidth costs (p2p communications)
- C_o - overhead costs (config, updates, power, uptime maintenance, monitoring)

so that

$$C_{tot} = N \times \widetilde{C_n} = N \times (C_c + C_s + C_r + C_b + C_o) \tag{3}$$

Since 2010 bitcoin protocol software improvements have seen almost 2 orders of magnitude cost savings in the validating and storage of transactions and blocks, i.e. C_c and C_s. For domestic sites, technical improvements in hardware are expected to keep the average costs for these components affordable for the foreseeable future. However, from rough estimates, the bandwidth costs appear to be a near term bottleneck so it is worth modelling in some more detail. Note also that a similar first order modelling approach can be probably used for other component costs.

2.2. **Bandwidth costs.** For a simplified approximation we assume that the cost of bandwidth is some function of time, t, and transaction volume rate, $\dot{\tau_x}$,

$$C_b \sim C_b(t, \dot{\tau_x}) \tag{4}$$

where if $\dot{\tau_x}$ is constant then

$$C_b \sim e^{-bt} \tag{5}$$

where b is the decay factor of bandwidth cost. Similarly supposing bandwidth costs per transaction are held constant but $\dot{\tau_x}$ grows exponentially then

$$C_b \sim \dot{\tau_x} \sim e^{\tau t} \tag{6}$$

where τ is the growth factor. In the case of both parts varying, i.e. bandwidth costs per transaction exponentially decrease while the volume rate of transactions grows exponentially we get

$$C_b = C_{b0} e^{(\tau - b)t} \tag{7}$$

C_{b0} being the cost of bandwidth at $t = t_0$, such that to avoid runaway node bandwidth costs

$$\tau \leq b. \tag{8}$$

Heuristics for b? Some estimates by CISCO have placed bandwidth technology improvements over longer time scales at around a doubling every 4 years. For example, a 1Mbyte block limit in 2010 would be equivalent to 2Mbyte in 2014, 4Mbyte in 2018 and so on, eventually hitting 32Mbyte in 2030, assuming the block limit was implicitly due to bandwidth limitations.