Ethereum Rust : PoL and Memo featured Erc-20 token.

Yasin Aktimur
yasak@gmx.com
ethereumrush.org
18 oct. 2019

Özet. Zaman içinde gördüğümüz kadarı ile proof of stake konsensus mekanizması yani bir kriptoparayı kiralayarak belirli bir süre sonunda ödül kazanma diğer bir deyiş ile masternode fikri en az iş kanıdı (Proof of work) konsensus algoritması kadar iyi iş başardı. Fakat insanlar bir coinin masternode'u olabilmek için gereğinden fazla programlama ve terminal bilgisi öğrenmek zorunda kalıyorlar bu da katılımı ciddi oranda azaltıyor. Birinci çözümümüz masternode olmayı teknik bilgi olmadan yapabilmeyi sağlayıp bunu yaparken de cihazlarını açık tutarak iş kanıt ispatı yapmalarına da devam etmelerini sağlamak. Diğer önemli çözümümüz de zaten eos ve stellar gibi kriptoparalarda var olan memo yani paranın yanında bir text (alfanümerik karakterler) göndermeyi ethereum ağında da aktif hale getirmek. Bunun neden kritik bir özellik olduğunu aşağıda açıklayacağım.

1- Memo özelliği.

Daha önce bir kriptopara borsasına üye olduysanız her üye için farklı bir adres yaratıldığını fark etmiş olmalısınız hatta bazı borsalar her işlem için farklı adres yaratıyorlar. Bunun bir sebebi var alice ve bob adında iki farklı üye hayal edin bir de borsaya ait bitcoin cüzdanı olsun ve üyelerden birisi bu cüzdana kendi hesabına yüklenmesi için 5 bitcoin gönderiyor borsanın sahibi bu paranın alice'den mi yoksa bob'dan mı geldiğini nasıl bilecek? Bu yüzden alice'e ayrı bob'a ayrı bir adres veriyor alice'in adresi 11a bob'un adresi 22b ile bitiyor olsun 11b adresine 5 bitcoin geldiği zaman borsa bu paranın bob tarafından gönderildiğini biliyor ve bob'a 5 bitcoinini site üzerinde yansıtmaya başlıyor fakat burada üyelerin bilmediği ama borsanın yaşadığı bir sorun var borsa gelen bu parayı kendi ana cüzdanına aktarmak zorunda bu yüzden bob'un bitcoin adresinden kendi adresine bir işlem daha gerçekleştiriyor ve gereksiz yere madencilere bir işlem ücreti ödemek zorunda kalıyor. Sadece iki üyesi olsa belki sorun yok ama günlük 50.000 yada 700 işlem gerçekleştirdiğinizi düşünün paraları tek bir adreste toplamak ayrı bir sorun teşkil ediyor.

Şimdi birde bizim alternatifimize göz atalım borsaya ait 000 ile biten bir bitcoin cüzdanı hayal edelim. Bob bu cüzdana para gönderirken para miktarını yanında da bob olduğunu kanıtlayan bbb alfanümeriğini de gönderdiğini varsayalım. Alice'de aynı cüzdana para göndereceği zaman miktar ve yanında aaa gönderiyor olsun. Paralar tek bir hesapta temiz bir şekilde birikiyor ve işletmeci ödemelerin kimlerden geldiğini rahatlıkla ayırt edebiliyor işte bu tasarruf ve verimlilik sağlıyor.

İşler böyle yürüdüğünde bir kârımız daha oluyor birinci senaryoda üye iş yerinin kriptopara ödemesi alabilmesi için bu konuda detaylı bir bilgiye sahip olması sıfırdan cüzdan yaratmayı bilmesi gibi bir çok farklı konuya hakim olması gerekirken ikinci senaryodaki kişi hiç bir şey yapmasına gerek kalmıyor bu da adaptasyonu arttıran ve coin'in talebini arttıracak bir özellik.

2- PoL: Proof of Live ~ Konsensus Detayları.

PoL : Proof of live yani canlı veya online kalma kanıdı konsensus algoritması masternode'ların ödüllerini cihazlarını açık tuttuğu sürece aldıkları aksi taktirde ödül toplayamadıkları bir sistemdir. Peki cihazlar online olduklarını diğer kullanıcılara merkeziyetsiz bir şekilde nasıl kanıtlayabilir.

Ethereum ağında her 14 ~ 15 saniyede bir yeni blok üretilir. Bu, bitcoin için 10 dakikadır. Ethereumdaki her blok aşağıdaki detaylara sahiptir. Ethereum'daki #8801692. blok üzerinden konuşmamız gerekirse bu blok başlığında;
Block Height: 8801692
Timestamp : X days x hrs ago (Oct-24-2019 07:34:43 AM +UTC)
Nonce : 0x5ae4dca59e388a83
gibi bir çok farklı değer bulunur ve bu değerler her 12 saniyede bir değişir örneğin nonce değeri madencilerin uğraşarak buldukları ve ödül almalarını sağlayan rastsal bir değerdir.
https://etherscan.io/block/8801692 etherscan üzerinden tüm blokları inceleyerek bu detayları görebilirsiniz.

Bu bloktaki nonce değeri rastlantısaldır ve kullanıcılar ödeme alması için sadece o blok geçerli iken yani o anki 12 saniye içerisinde hexadecimal yani 16 lık sayı sisteminde 0x5ae4dca59e388a83 olan sayıyı önce decimal yani 10 luk sisteme yani bizlerin kullandığı sayı sistemine çevirirler. Bu da 6549602361985763971 rakamına eşittir hexadecimal to decimal yazarak internetten online kendiniz de bu rakamları birbirine çevirebilirsiniz. Şimdi bu basitçe eğer nonce değerinin decimal karşılığının son rakamı 1 e eşitse ödülü dağıt deseydik şöyle bir sonuç alacaktık. Her 12 saniyede bir 0 ile 9 arasında bir rakam elde edeceğimiz için ortamla 9 + 1 blokta bir ödül dağıtacaktık [+1 sıfır'ı temsil ediyor] bu da her 120 saniyede bir anlamına geliyor yani 2 dakikada bir. Fakat biz ödemeleri günlük yapmak istiyoruz. Bir gün toplamda 86.400 saniyedir. Bu sayıyı 15 'e böldüğümüzde 5.760 rakamına ulaşırız bu da bize bir günde üretilen ortalama ethereum blok sayısını verir. Biz 5760 blokta bir yani ortalama günde bir ödül vermek istiyorsak nonce değerinin Modüler arimatikteki mod 5760 (x% 5760) almamız gerekiyor. Bu arada yukarıda sayının son hanesini almakta mod 10 anlamına geliyor yani sayıyı sürekli 10 bölüyoruz ve en son kalan mod 10 anlamına geliyor mesela 101 sayısının mod 10 u 1 dir. İşte bir sayının mod 5760 aldığımızda bu rastgele sayıdan 0 ile 5760 - 1 arasında sayılar elde ediyoruz. Böylece ödülü bir günde bir dağıtıyoruz ama ödülün hangi blokta verileceğini kainattaki hiç kimse bilmiyor ve her 12 saniyede bir kontrol etmesi gerekiyor. Bu da Proof of Live algoritmasının nasıl çalıştığını açıklıyor.

Aşağıdaki python kodları ile mod 5760 'ın gerçekten'de 0 ile 5759 arasında rastgele rakamlar ürettiğini kendiniz de görebilirsiniz ayrıca her üretilen bu rastgele değerlerleri perfecttiming kümesinde biriktirip en son bu sayıların artirmatik ortalamasını aldığımız zaman'da 5760 'a çok yakın bir değer elde ediyoruz.

```
import random
perfecttiming = []
totalrty = 0
for luckynumber in range(0,1000000):
    nonce = random.randint(10000000, 10000000000)  # 10 milyon ile
10 milyar arasında rastgele sayı üretir.
    y = nonce % 5760
    if y == 0:
        perfecttiming.append(totalrty)
        totalrty = 0
        continue
    else:
        totalrty = totalrty + 1
        continue

print("perfecttiming: ", perfecttiming)
print("perfecttiming len: ",len(perfecttiming))
a = 0
for i in perfecttiming:
    a = a + i

print(a/len(perfecttiming))
```

```
perfecttiming:  [5471, 1352, 4305, 3153, 9511, 4122, 6158, 2417,
892, 2907, 2657, 2169, 1521, 4039, 3082, 14402, 17539, 5964,
12465, 3894, 32175, 1629, 731, 7506, 1301, 5530, 5785, 8534, 4267,
2225, 2496, 7074, 7474, 13839, 9079, 3540, 1702, 189, 2383, 4696,
1526, 826, 12249, 11315, 18121, 2352, 1465, 5799, 1425, 9538,
5226, 6980, 3175, 13353, 2567, 847, 16470, 4176, 313, 5236, 90,
12461, 4183, 6848, 26720, 8427, 2250, 8434, 1292, 3310, 1933,
5939, 1814, 2328, 8950, 1138, 264, 2739, 1704, 7180, 1478, 4724,
8951, 11182, 617, 1378, 182, 3475, 4224, 10214, 20975, 686, 4600,
2518, 13435, 8601, 6234, 384, 6575, 3069, 5343, 1623, 3555, 529,
353, 7958, 3435, 8727, 809, 625, 12863, 2664, 865, 5006, 4841,
2178, 3991, 6878, 10098, 341, 3537, 2226, 1048, 21701, 4228,
12542, 3904, 12259, 415, 2617, 2228, 8, 2919, 6103, 7354, 902,
737, 575, 6538, 1356, 2632, 11370, 13193, 13159, 2210, 10627,
4762, 16088, 5594, 3524, 15666, 14255, 4919, 896, 50725, 6094,
2838, 103, 2881, 15713, 14489, 4145, 1751, 1099, 1449, 291, 11154,
1198, 787, 3361, 1053, 2346, 5964]
perfecttiming len:  173
5762.167630057804
```

Ethereum Rust Ödül dağıtım algoritması

PoL konsensus algoritmasında masternode için kilitli kalması gereken miktar kullanıcılar tarafından belirlenir ve ismi maximum target'dir ve bir kullanıcı masternode olmak için 3 aylığına 50.000 coini kilitlediği taktirde maximum target 50.000 olarak belirlenir ve diğer kullanıcıların masternode olabilmesi için en az maximum target'in %1 i kadar ödeme yapmaları gerekir. 50.000 örneğinde gerekli minimum rakam 500 dür. Maximum target'in %1 i kadar yatırım yapan bir yatırımcı maximum target bedelini ödeyen kullanıcıdan %9 oranında daha az kazanç sağlar. En yüksek kazancı maximum target'i ödeyen kişi alır ve her %10 luk dilimde kazanç oranı %1 ilk düşüş gösterir. Bu örnekte ben 25.000 coin kilitlediğimde %50 ilk bant üzerinden maximum targettaki kullanıcıdan %5 daha az ödül ödemesi alırım.

Daha fazla kilitleyen kişinin daha fazla ödeme alması konsepti PoL tabanlı kripto paraların alım için bir talep yaratır, böyle bir durumda insanlar daha fazla kazanmak için piyasadan daha fazla kripto para toplamak zorunda kalacaktır bu da piyadaki arzı düşürerek paranın sürekli değerli kalmasını sağlar.

Ethereum Rust arz

Toplam arz 22 milyon ile sınırlıdır ve ilk 1 milyonu önceden kazılmıştır. Bu kazılan coinler ile masternode'lar kazanç sağlamaya ve coin'i üretmeye devam ederler.
Bir günün toplamda 86.400 sn olduğunu ve bu sayıyı 15 'e böldüğümüzde 5.760 rakamına ulaştığımız hatırlayın Bitcoin'de block ödül süresi 10 dakikada bir dir.Bir gün 1440 dakika olduğu için bitcoin 'de günde ortalama 144 adet block üretir. 4 senede 365*4 ten 1460 adet gün olduğu için 1460 * 144 = 210.240 'a eşittir bu da bitcoin'in neden 210.000 blok yani 4 senede bir yarılandığını açıklar. Ethereum Rust arz dengesinde nakamoto consensus protokolünün güncellenmiş bir versiyonunu destekler. ETR premised 1 milyondan sonra 21 milyon adet üretilecektir ama yarılanmanın 2 senede bir olmasını uygun görüyoruz bu yüzden blok yarılanmaları 210.000 blokta bir değilde 105.000 blokta bir olmakta. Onun dışında bitcoinde ödüller 10 dakikada bir 50 adet dağıtılıyor. Ethereum ağında kullanıcılar ödüllerini almak için bir fonksiyon çalıştırmak zorunda ve bu fonksiyonda ethereum ile ödeme yapılarak çalışıyor bu yüzden kullanıcılara 10 dakikada bir ödeme yaptırma fikri kullanışsız bu yüzden ödemeler 10 dakikada 50 yerine 144*50 üzerinden 7200 adet olarak toplu bir ödeme yapılır. İşte bu 7200 adet ödül o anda masternodelar arasında ethereum rust ödül dağıtım algoritmasına göre dağıtılır.