Ouroboros Leios: Mục tiêu và ý tưởng thiết kế

MỘT NGHIÊN CỨU THẢO LUẬN CỦA IOG

Duncan Coutts duncan@well-typed.com duncan.coutts@iohk.io
Giorgos Panagiotakos giorgos.panagiotakos@iohk.io
Matthias Fitzi matthias.fitzi@iohk.io

Phiên bản 1.0, tháng 11 năm 2022

Mục đích của Ouroboros Leios

Động lực cho Ouroboros Leios – một biến thể mới của gia đình Ouroboros – là tăng đáng kể thông lượng, đồng thời đạt được ít nhất các đặc tính bảo mật tốt như các biến thể Ouroboros trước đây.

Các biến thể hiện có của thuật toán Blockchain Ouroboros bị giới hạn về thông lượng mà chúng có thể đạt được – cả thông lượng dữ liệu và thông lượng xử lý CPU. Chúng không bị giới hạn chủ yếu bởi các tài nguyên có sẵn cho mỗi Node (dung lượng mạng hoặc hiệu suất CPU), mà bởi bản chất của các phụ thuộc dữ liệu và phụ thuộc giao tiếp trong thuật toán phân tán. Cải thiện điều này đòi hỏi một thiết kế thuật toán mới – đó là mục tiêu của Ouroboros Leios.

Ngoài ra, một thiết kế mới mang đến cơ hội kết hợp các tính năng hiện đại hữu ích khác: phí giao dịch theo cấp độ với mức độ ưu tiên dịch vụ tương ứng và đồng bộ hóa chuỗi nhanh hơn bằng cách loại bỏ nhu cầu thực hiện mọi hợp đồng thông minh.

Tất nhiên, có sự đánh đổi trong thiết kế, đặc biệt là tăng sử dụng tài nguyên, tăng độ trễ giao dịch và những điều sẽ được thảo luận.

Tuy nhiên, thiết kế Ouroboros Leios mới không phải là một phần mở rộng nhỏ hoặc mô đun. Nó là một phần mở rộng đáng kể của các thiết kế Ouroboros Praos và Genesis, và những thay đổi đối với việc triển khai thực tế cũng sẽ rất đáng kể.

Đối tượng dự định

Đối tượng dự định của nghiên cứu thảo luận này là bất kỳ ai trong cộng đồng Cardano quan tâm đến hiệu suất của Cardano hoặc sự phát triển tiềm năng trong tương lai của Cardano. Cuộc thảo luận có phần kỹ thuật, vì vậy những người tò mò về kỹ thuật sẽ tận dụng tối đa nó.

Nội dung

1 Giới hạn thông lượng hiện có

Một đặc điểm của tài nguyên mạng và CPU là "sử dụng hoặc mất nó": không bao giờ lấy lại được thời gian không sử dụng các tài nguyên có sẵn. Do đó, việc sử dụng các tài nguyên này theo kiểu "tăng vọt" (Spiky) sẽ khiến các tài nguyên không được sử dụng đúng mức.

Trong biến thể Ouroboros được triển khai hiện tại – Ouroboros Praos – chúng ta có thể quan sát thấy hai cách chính mà thuật toán không sử dụng hết tài nguyên mạng và máy tính có sẵn trên mỗi Node.

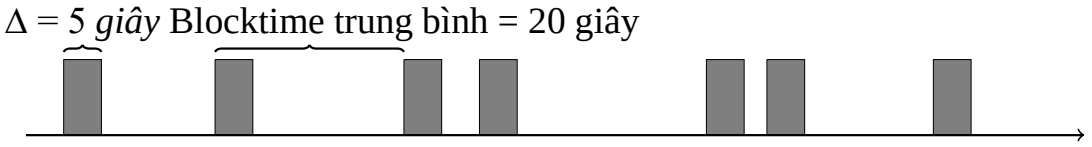
1.1 Block khuếch tán là một phần của tổng thời gian khối (Block time)

Trong Praos, thời gian mà một Block được tạo ra và phân tán trên mạng lưới chỉ là một phần nhỏ so với tổng thời gian trung bình giữa các Block. Ví dụ: với các tham số cho mạng chính Cardano: các Block được tạo trung bình cứ sau 20 giây, nhưng thời gian để phân tán các Block trên toàn mạng lưới dự kiến sẽ duy trì trong vòng 5 giây (đây là tham số Δ). Điều này có nghĩa là trung bình ba phần tư thời gian được dành cho việc nhàn rỗi. Xem Hình 1 để biết minh họa.

Điều này không thể dễ dàng thay đổi bằng cách điều chỉnh các tham số: đối số bảo mật cho Praos dựa trên thời gian khuếch tán (Δ) là một phần nhỏ của thời gian trung bình giữa các Block. Theo trực giác, điều này là do nếu thời gian khuếch tán trở thành một phần lớn của thời gian khối trung bình thì sẽ có tỷ lệ "trận chiến khối" (Block Battles) ngày càng cao hơn, điều này sẽ ảnh hưởng đến lợi nhuận bảo mật của giao thức. Ví dụ: nếu giảm thời gian giữa các Block thì để duy trì bảo mật hệ thống ở cùng mức, thời gian khuếch tán (Δ) cho phép cũng sẽ phải giảm. Điều này chỉ có thể được giảm bớt bằng cách giảm kích thước của các Block hoặc bằng cách giảm ngân sách thời gian cho mỗi Block để thực thi các tập lệnh trong giao dịch. Hiệu quả tổng thể sẽ là giảm thông lượng, thậm chí có tính đến các Block thường xuyên hơn.

Lưu ý rằng nếu lịch trình của Slot Leader là công khai thay vì riêng tư thì có thể đóng gói dày đặc hơn các khoảng thời gian khuếch tán: dày đặc hơn 4 lần. Đây là cái giá mà chúng tôi phải trả để có được sự bảo mật bổ sung của một lịch trình Slot Leader riêng tư trong một Blockchain tuyến tính đơn giản truyền thống.

Ví dụ lịch sản xuất Block Praos ngẫu nhiên

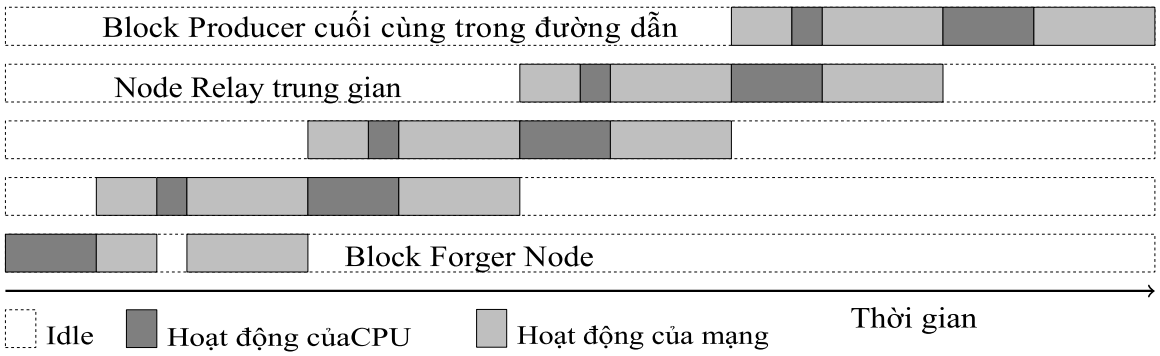


Hình 1: Thời gian khuếch tán (Δ) của Block Praos là một phần thời gian trung bình giữa các Block

1.2 Block khuếch tán không sử dụng hết tài nguyên

Cách thứ hai quan trọng hơn, các tài nguyên có sẵn không được sử dụng là trong chính quá trình khuếch tán Block: tại bất kỳ thời điểm nào trong khi Block mới lan truyền khắp biểu đồ mạng lưới, chỉ các Node trên "mặt sóng thông tin" (Information Wavefront) đang hoạt động và sử dụng tài nguyên mạng và tính toán. Nói cách khác: trong quá trình khuếch tán Block, trước khi Block đến một Node thì Node đó ở trạng thái không hoạt động (Idle) và sau khi đã tải xuống, xác thực và chuyển tiếp Block thì Node đó sẽ trở lại trạng thái không hoạt động[1].

Xem Hình 2 để minh họa. Lưu ý phần lớn thời gian cho hầu hết các Node được sử dụng là không hoạt động. Một điều không rõ ràng ngay lập tức từ hình minh họa là cũng có tương đối ít chồng chéo giữa hoạt động của CPU và hoạt động mạng trên mỗi Node. Điều này có nghĩa là việc sử dụng từng tài nguyên thậm chí còn ít hơn so với lần đầu tiên xuất hiện.



Hình 2: Hoạt động và không hoạt động của Node trong quá trình chuyển tiếp Block đường ống giữa 5 Node

[1]Có thêm công việc sau khi áp dụng một Block để cập nhật và nạp lại Mempool, công việc này có cùng mức độ quan trọng của công việc như chính việc xác thực Block.

1.2.1 Định lượng việc sử dụng tài nguyên CPU

Hãy xem xét rằng việc xác thực một Block hiện dự kiến sẽ mất khoảng 50-100 Mili giây. Vì vậy, tốt nhất đối với mỗi Node, đây chỉ là vài trăm Mili giây trong số 5 giây cho thời gian khuếch tán tổng thể Δ.

Chi phí này gần như tăng gấp đôi khi xác thực các giao dịch vào Mempool. Chi phí này không nằm trong đường tới hạn khuếch tán và có thể được phân bổ theo thời gian. Có những chi phí và chi phí chung khác, nhưng điều này cho thấy mức độ quan trọng liên quan.

1.2.2 Định lượng việc sử dụng tài nguyên mạng

Tình hình sử dụng tài nguyên mạng phức tạp hơn. Có 2 đặc điểm quan trọng ảnh hưởng đến việc sử dụng tài nguyên mạng.

1. Các Node không chỉ gửi các Block tới một Node ngang hàng xuôi dòng (Downstream Peer), chúng có nhiều Node ngang hàng xuôi dòng. Điều này làm tăng việc sử dụng tài nguyên mạng tỷ lệ thuận với số lượng Node ngang hàng xuôi dòng.
2. Việc truyền một Block giữa 2 Node ngang hàng chỉ sử dụng tài nguyên mạng trong thời gian ngắn: một lần trên thiết bị gửi để "tuần tự hoá" Block vào dây và một lần nữa trên thiết bị nhận để "gỡ tuần tự hóa" Block từ dây. Không Node ngang hàng nào sử dụng tài nguyên mạng cục bộ trong khi Block "đang được truyền" (In Flight) giữa chúng.

Để định lượng các hiệu ứng này, hãy xem xét việc gửi một Block 100kB qua liên kết mạng 1.000.000kB/giây[2] với độ trễ 100 Mili giây: đầu gửi sẽ dành 0,1 Mili giây để tuần tự hoá Block lên dây và sau 100 Mili giây, đầu nhận cũng sẽ dành 0,1 Mili giây để gỡ tuần tự Block ra khỏi dây. Nếu cùng một Node ngang hàng có 1.000 Node xuôi dòng thì về tổng thể, nó sẽ giữ cho giao diện mạng cục bộ của nó bận trong 100 Mili giây trong tổng thời gian khuếch tán Δ 5000 Mili giây.

Một lần nữa, có những chi phí chung nhưng điều này đưa ra một dấu hiệu về thứ tự cường độ liên quan.

2 Mục tiêu thiết kế

2.1 Thông lượng

Mục tiêu thiết kế chính là tăng thông lượng Blockchain bằng cách sử dụng một phần đáng kể băng thông mạng và sức mạnh CPU của mỗi Node và làm như vậy trên cơ sở cơ bản liên tục.

[2]Tốc độ này xấp xỉ 10GBit/giây, nhưng làm cho các con số trong ví dụ rõ ràng hơn.

Có hai biện pháp chính và một biện pháp dẫn xuất của thông lượng Blockchain mà chúng tôi muốn tăng:

1. Thông lượng dữ liệu được đo bằng Byte trên giây (thường là Kilobyte trên giây, kB/s);
2. Thông lượng tập lệnh được đo bằng giây CPU trên mỗi thời gian đồng hồ treo tường tính bằng giây[3] (thường là Mili giây CPU trên giây đồng hồ treo tường, ms/s);
3. Thông lượng giao dịch được đo bằng giao dịch mỗi giây (TPS). Đây không phải là một biện pháp độc lập mà được bắt nguồn từ hai biện pháp kia. Xem Phần 2.2 để biết phần thảo luận về TPS như một biện pháp.

Để minh họa hai biện pháp chính này, chúng ta hãy xem xét chúng là gì đối với mạng chính Cardano hiện tại. Tại thời điểm viết bài, các Block mạng chính của Cardano được phép có dung lượng tối đa là 88kB và chúng được tạo ra trung bình cứ sau 20 giây. Vì vậy, thông lượng dữ liệu hiện tại là 88kB/20s = 4,4kB/s.

Để so sánh với các tài nguyên có thể có, hãy xem xét một máy chủ từ trung cấp đến cao cấp trong trung tâm dữ liệu sẽ có giao diện mạng 10Gbit/s. Nếu máy chủ này chia băng thông của nó cho 1000 Node ngang hàng (và chúng tôi cho phép khoảng 20% chi phí hoạt động) thì điều này cung cấp khoảng 900kB/giây băng thông hữu ích cho mỗi Node ngang hàng. Hoặc xem xét rằng băng thông rộng tại nhà 10Mbit/s cũng cung cấp khoảng 900kB/s băng thông hữu ích (với chi phí chung tương tự).

Đối với thông lượng tập lệnh,tại thời điểm viết bài, các Block mạng chính Cardano được phép sử dụng tối đa[4] 40 Mili giây danh nghĩa cho mỗi Block. Vì vậy, thông lượng tập lệnh hiện tại là 40ms/20s = 2ms/s.

Để so sánh với thông lượng tập lệnh là 2 ms/s, hãy xem xét một CPU Core đơn lẻ chạy liên tục có sẵn thời gian xử lý 1000 ms/s. Nếu nhiều Core khả dụng và có thể được sử dụng thì số giây CPU khả dụng trên mỗi giây đồng hồ treo tường sẽ lớn hơn 1.

Cần lưu ý rằng việc cố gắng sử dụng tất cả tài nguyên mạng và CPU mọi lúc là không hợp lý. Các thuật toán Blockchain trong gia đình Ouroboros dựa vào khả năng chuyển đổi nhánh, liên quan đến việc "bắt kịp" nhánh đó và điều quan trọng là có thể bắt kịp tương đối nhanh chóng – chắc chắn là nhanh hơn thời gian thực. Hơn nữa, vì Ouroboros sử dụng phân phối Poisson ngẫu nhiên cho lịch trình sản xuất Block, nên đương nhiên sẽ có những đợt công việc bùng nổ và tạm lắng. Điều này đặt ra giới hạn về phần tài nguyên nào có thể được sử dụng trong hoạt động

[3]Đơn vị là không thứ nguyên, là giây trên giây. Tuy nhiên, chúng tôi sẽ sử dụng đơn vị "ms/s" vì giây đo lường những thứ khác nhau: thời lượng hoạt động của CPU so với thời lượng của đồng hồ treo tường.

[4]Như đã hiệu chuẩn trên máy tham chiếu. Thời gian thực tế sẽ thay đổi.

bình thường, để việc bắt kịp có thể được thực hiện đủ nhanh và có thể xử lý các cụm ngẫu nhiên.

Xem thêm Phần 8 để biết phần thảo luận về tốc độ và mức độ ngốn tài nguyên mà chúng tôi có thể muốn cho mạng chính Cardano.

2.2 TPS

Giao dịch mỗi giây (TPS) là thước đo thông lượng thường được sử dụng để so sánh các loại Blockchain khác nhau (và các hệ thống xử lý dữ liệu Non-Blockchain khác). TPS là thước đo phái sinh dựa trên thông lượng dữ liệu, thông lượng thời gian thực thi tập lệnh và quy mô của giao dịch. Vì lý do này, chúng tôi tập trung vào các phép đo cơ bản của dữ liệu và thông lượng thực thi tập lệnh. Kết quả của TPS có thể được tính toán.

Vì vậy, mặc dù mục tiêu của Ouroboros Leios là tăng đáng kể thông lượng được đo bằng TPS, nhưng đây sẽ là kết quả của việc tăng thông lượng thực thi tập lệnh và dữ liệu như đã thảo luận trong phần trước.

Cũng cần lưu ý rằng TPS là thước đo gây khó khăn trong việc so sánh công bằng giữa các hệ thống. Điều này là do nó phụ thuộc chủ yếu vào quy mô của giao dịch và thời gian chạy bất kỳ tập lệnh nào trong đó. Cách đơn giản để làm cho một hệ thống trông đẹp mắt trên các biện pháp TPS là sử dụng quy mô giao dịch càng nhỏ càng tốt, không có tập lệnh (hoặc rất đơn giản).

Ví dụ: giả sử một Blockchain có thông lượng dữ liệu là 500 kB/s. Với các giao dịch 512 Byte, điều này sẽ mang lại 1000 TPS. Đối với cùng một thông lượng dữ liệu nhưng với các giao dịch đều tải lên các tập lệnh khổng lồ 16kB, kết quả sẽ chỉ là 31 TPS.

2.3 Độ trễ

Trong thiết kế hệ thống xử lý dữ liệu thường xảy ra trường hợp đánh đổi giữa thông lượng và độ trễ. Các thiết kế xử lý nhiều giao dịch hơn mỗi giây thường làm như vậy với chi phí tăng độ trễ cho các giao dịch riêng lẻ.

Trong trường hợp hệ thống Blockchain, định nghĩa thông thường về độ trễ giao dịch là khoảng thời gian từ khi người dùng gửi giao dịch vào hệ thống đến khi giao dịch đó được đưa vào một Block có sẵn cho hầu hết[5] người dùng khác. Lưu ý rằng đây không phải là thời điểm để giao dịch trong một Block ổn định với xác suất cao.

Đối với thiết kế Ouroboros Leios, chúng tôi sẵn sàng hy sinh hợp lý độ trễ giao dịch để đạt được sự cải thiện đáng kể về thông lượng.

[5]Chính xác hơn là sẽ có độ trễ để tiếp cận các tỷ lệ người dùng khác nhau, chẳng hạn như 95% hoặc 99%.

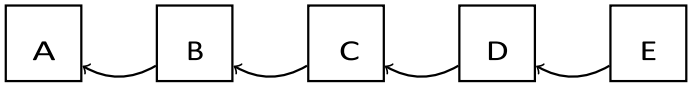
2.4 Bảo mật

Một số hệ thống Crypto thông lượng cao hiện đại thực hiện các thỏa hiệp bảo mật để đạt được thông lượng cao, chẳng hạn như chỉ chống lại 30% đối thủ. Để so sánh, các phiên bản hiện tại của Ouroboros có khả năng chống lại đối thủ chiếm tới 50% cổ phần, mặc dù có thông lượng tương đối thấp.

Mục tiêu của Ouroboros Leios là đạt được thông lượng cao mà không ảnh hưởng đến bảo mật: để đạt được các thuộc tính bảo mật chính thức giống hoặc tốt hơn như các phiên bản trước của Ouroboros.

3 Chiến lược thiết kế

Như đã thảo luận trong Phần 1.2, trong Ouroboros Praos, các Block khuếch tán qua mạng chỉ sử dụng một phần tài nguyên mạng và CPU của mỗi Node. Lý do cơ bản cho điều này là một hệ thống các Node phân tán là một hệ thống *song song*, nhưng thuật toán Blockchain Ouroboros Praos là một thuật toán *tuần tự* (hầu hết). Bất kỳ thuật toán tuần tự nào được thực hiện trên một hệ thống song song nhất thiết sẽ để lại các tài nguyên không được sử dụng. Thuật toán để xây dựng Blockchain nhất thiết phải tuần tự vì bản thân cấu trúc của Blockchain là tuyến tính. Nó là tuyến tính theo nghĩa là các phụ thuộc dữ liệu trong một chuỗi hợp lệ là tuyến tính: mỗi Block phụ thuộc vào Block trước đó. Xem Hình 3 để được minh họa. Cụ thể, việc xác thực một Block phụ thuộc vào việc tính toán trạng thái sổ cái cho Block, được tính như một chức năng của trạng thái sổ cái cho Block trước đó. Như vậy có sự phụ thuộc dữ liệu trực tiếp giữa các Block liên tiếp và về tổng thể, điều này tạo thành một chuỗi phụ thuộc dữ liệu tuyến tính.



Hình 3: Một Blockchain truyền thống với các phụ thuộc dữ liệu tuyến tính của nó. Ví dụ: Block B *phụ thuộc vào* Block A: trạng thái sổ cái của B được tính từ trạng thái sổ cái của A

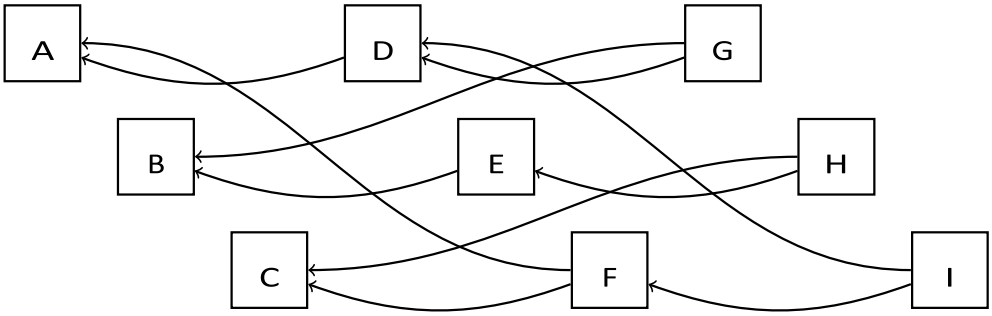
3.1 Một Blockchain đồng thời và một thuật toán song song

Dựa trên quan sát này, chúng ta có thể thấy rằng nếu có một "Blockchain đồng thời" với đủ phụ thuộc dữ liệu *đồng thời*, chúng ta có thể tìm thấy một thuật toán phân tán *song song*[6] để xây dựng chuỗi. Nếu chúng ta có thể kiểm soát mức độ đồng thời trong cấu trúc của chuỗi và mức độ song song trong thuật toán, thì điều

[6]Chúng tôi phân biệt đồng thời và song song: đồng thời là về dữ liệu hoặc sự kiện không được sắp xếp theo thứ tự đối với nhau, trong khi song song là sử dụng nhiều phần cứng máy tính hơn để tính toán nhanh hơn.

này có thể cho phép chúng ta tăng khối lượng công việc đến mức chúng ta có thể sử dụng một tỷ lệ đáng kể các nguồn lực sẵn có.

Đây chính xác là những gì Ouroboros Leios cố gắng thực hiện.



Hình 4: Một mô hình tưởng tượng về sự phụ thuộc dữ liệu đồng thời giữa các Block. Ví dụ, khối F phụ thuộc vào khối A và C, nhưng khối D, E và F không có sự phụ thuộc giữa chúng.

Xem Hình 4 để biết minh họa về một mô hình phụ thuộc dữ liệu tưởng tượng giữa các Block. Ở ví dụ trong hình minh họa, có tối đa 3 Block đồng thời. Người ta có thể tưởng tượng làm thế nào mô hình này có thể được mở rộng cho số lượng Block đồng thời cao hơn. Tính đồng thời trong cấu trúc càng cao thì cơ hội khai thác song song nhiều tài nguyên phần cứng càng lớn.

3.2 Sổ cái trên một Blockchain đồng thời

Cấu trúc Blockchain đồng thời có ý nghĩa quan trọng đối với sổ cái được xây dựng trên Blockchain đó. Sổ cái sẽ cần hỗ trợ cấu trúc Blockchain phi tuyến tính và điều này sẽ không thể thực hiện được đối với tất cả các sổ cái. Có một vài vấn đề quan trọng:

1. Điều quan trọng là phải hiểu ý nghĩa của sổ cái được hình thành từ việc kết hợp kết quả của nhiều Block đồng thời. Sự cố này phát sinh ở bất kỳ nơi nào có nhiều dòng phụ thuộc dữ liệu tham gia. Ví dụ trong Hình 4, Block F phụ thuộc vào Block A và C, chúng đồng quy với nhau. Nếu các giao dịch không tương thích lẫn nhau được bao gồm trong các Block đồng thời thì việc giải thích sổ cái kết quả là gì?
2. Trong trường hợp một sổ cái được hình thành từ các Block đồng thời được nối sau đó, chi phí tính toán cần phát sinh chủ yếu với chính các Block đồng thời chứ không phải với điểm mà chúng nối với nhau. Nếu không, hầu hết lợi thế song song của chuỗi đồng thời sẽ bị mất.

Đây không phải là những vấn đề đơn giản. Các giao dịch trong sổ cái có thể phụ thuộc hoặc xung đột với nhau. Theo định nghĩa, các Block đồng thời không thể

phụ thuộc vào nhau. Vì vậy, thách thức là phải làm gì với các giao dịch trong các Block đồng thời sẽ phụ thuộc hoặc xung đột với nhau.

3.2.1 Hầu hết các giao dịch độc lập

Mặt khác, cơ hội phát sinh từ thực tế là *hầu hết* các giao dịch "đang xử lý" cùng một lúc độc lập với nhau và các giao dịch độc lập có thể được xử lý đồng thời. Thật vậy, giới hạn cuối cùng về mức độ đồng thời trong cấu trúc Blockchain là mức độ mà các giao dịch đang xử lý là độc lập. Tất nhiên, sự kết hợp giữa phụ thuộc hoặc độc lập trong các giao dịch đang xử lý phụ thuộc vào cách sử dụng sổ cái. Người ta có thể tưởng tượng một sổ cái chuyên dụng quản lý một tài nguyên dùng chung trong đó hầu hết các giao dịch phụ thuộc hoặc xung đột với nhau, nhưng đối với một sổ cái mục đích chung (chẳng hạn như nền tảng "Layer 1"), kỳ vọng hợp lý là hầu hết các giao dịch đang xử lý đều độc lập.

Một giải pháp không phải là trì hoãn việc xử lý các giao dịch đến điểm mà sổ cái từ các Block đồng thời được kết hợp, sau đó xử lý các giao dịch theo thứ tự tuyến tính có thể giải quyết các quan hệ phụ thuộc của chúng. Đây không phải là giải pháp vì nó lãng phí tính song song của CPU có sẵn và sẽ không thể tận dụng hết thời gian của CPU.

3.2.2 Tính tuần tự

"Tiêu chuẩn vàng" trong xử lý giao dịch đồng thời (ví dụ: trong cơ sở dữ liệu tập trung hoặc phân tán[7]) là "tính tuần tự". Việc xử lý đồng thời một tập hợp các giao dịch có thể được tuần tự hóa nếu kết quả của quá trình xử lý đồng thời giống *như khi* các giao dịch được xử lý nối tiếp theo một số thứ tự. Theo trực giác, thuộc tính này có nghĩa là chúng ta có một diễn giải rõ ràng về sổ cái: chúng ta đã hiểu ý nghĩa của việc xử lý một loạt các giao dịch theo thứ tự, và vì vậy nếu xử lý đồng thời cho cùng một kết quả thì sự hiểu biết đó được bảo toàn.

3.2.3 Kiểm soát tính đồng thời lạc quan

Đối với các sổ cái có mục đích chung, chúng ta có thể cho rằng hầu hết các giao dịch đang xử lý là độc lập, chúng ta có thể áp dụng phương pháp "lạc quan" để xử lý một số giao dịch hóa ra là xung đột:

1. Chúng ta xử lý đồng thời nhiều Block của giao dịch.
2. Trong mỗi Block, các giao dịch được xử lý theo thứ tự nên sự phụ thuộc giữa các giao dịch trong một Block có thể được xử lý bình thường.
3. Khi các kết quả được nối với nhau, các Block có thể sắp xếp theo thứ tự tuyến tính.

[7]Để đọc thêm hãy xem https://en.wikipedia.org/wiki/Serializability

4. Điều này làm phát sinh một thứ tự tuyến tính cho các giao dịch: do các Block được đưa vào một đơn đặt hàng và các giao dịch trong các Block đã được đặt hàng.
5. Bất kỳ giao dịch xung đột nào hiện có thể bị loại bỏ. Cụ thể, trong một tập hợp các giao dịch xung đột, giao dịch đầu tiên được giữ lại và phần còn lại bị loại bỏ.

Với cách tiếp cận này, các giao dịch xung đột sẽ làm giảm thông lượng hiệu quả, bởi vì chúng chỉ được xử lý để rồi bị loại bỏ sau đó. Tuy nhiên, nếu tỷ lệ xung đột không quá cao thì đây là một sự đánh đổi hợp lý. Lưu ý rằng điều quan trọng là phải tránh quá nhiều giao dịch xung đột nhân tạo với chính chúng bằng cách để cùng một giao dịch xảy ra trong các Block đồng thời.

Điều quan trọng là để duy trì tính song song, việc phát hiện và loại bỏ các giao dịch xung đột phải rẻ so với chi phí thực hiện các giao dịch (ví dụ: chạy tập lệnh và kiểm tra chữ ký mật mã).

3.2.4 Sổ cái kiểu UTxO

Rất may mắn, sổ cái kiểu UTxO được bố trí phù hợp để giải quyết những vấn đề này. Điều này là do thực tế là các giao dịch trong sổ cái UTxO xác định rõ ràng tất cả đầu vào và đầu ra của chúng ngay từ đầu, và những phần phụ thuộc đó đã hoàn tất. Chúng hoàn chỉnh theo nghĩa là không có "tác dụng phụ" nào khác của giao dịch ngoài các đầu vào và đầu ra được xác định rõ ràng. Điều này giúp dễ dàng xác định các giao dịch xung đột với nhau. Điều đó cũng có nghĩa là, với điều kiện là sự phụ thuộc giữa các giao dịch được tôn trọng, các giao dịch có thể được sắp xếp lại mà không làm thay đổi kết quả. Điều này giúp có thể thực hiện đúng quy trình tuần tự hoá nêu trên và thực hiện tương đối rẻ. Tính đúng đắn của quá trình này có thể được chính thức hóa bằng toán học.

Từ lâu, người ta đã quảng cáo rằng sổ cái UTxO có những lợi thế tiềm năng cho tính đồng thời, nhưng nó thường không được khai thác hiệu quả. Thật hài lòng khi thấy cách tiếp cận đồng thời này cuối cùng có thể khai thác những lợi thế đó.

4 Phác thảo thiết kế

Tất cả các phiên bản trước của Ouroboros đều sử dụng cấu trúc Blockchain tuyến tính truyền thống đơn giản. Ngược lại, Ouroboros Leios có cấu trúc Blockchain mới sáng tạo với *các phụ thuộc dữ liệu đồng thời*. Như đã thảo luận trong Phần 3, đây là chìa khóa để mở ra cơ hội tăng tính song song và sử dụng tài nguyên. Ouroboros Leios khai thác các phụ thuộc dữ liệu đồng thời trong cấu trúc Blockchain bằng cách sử dụng thuật toán *phân tán song song đồng thời* để xây dựng Blockchain.

Ngoài ra, Ouroboros Leios là một thuật toán *có thể mở rộng* theo nghĩa sau: mức độ phụ thuộc dữ liệu đồng thời trong cấu trúc Blockchain Leios có thể được kiểm soát, điều này ảnh hưởng trực tiếp đến mức độ song song có sẵn trong quá trình thực thi thuật toán phân tán để xây dựng nó, cũng như ảnh hưởng trực tiếp việc sử dụng tài nguyên và thông lượng.

4.1 Tổng quan về thuật toán

Có thuật ngữ mới cần lưu ý đối với các loại đối tượng mới liên quan đến cấu trúc và thuật toán Blockchain Leios: *khối đầu vào* (Input Block), *khối chứng thực* (Endorsement Block), *báo cáo chứng thực* (Endorsement Report), *chứng chỉ xác nhận* (Endorsement Certificate) và *khối xếp hạng* (Ranking Block). Tất cả sẽ được trình bày như một phần của tổng quan về thuật toán, tóm tắt trong Bảng 1 và chi tiết một lần nữa trong Phần 4.2.

4.1.1 Tạo Input Block

Các Node có thể được chọn (bằng xổ số VRF có trọng số cổ phần riêng tư) để tạo ra một Input Block. Khi làm như vậy, Node lấy một chuỗi các giao dịch từ Mempool hợp lệ đối với một điểm gần đây trên chuỗi hiện tại của Node (sẽ là một Ranking Block, xem Phần 4.1.6). Node lưu một tham chiếu đến cùng một điểm gần đây trên chuỗi. Tham chiếu sẽ được sử dụng sau này để đảm bảo rằng các Node khác sẽ có thể xác thực các giao dịch trong Input Block dựa trên cùng một trạng thái sổ cái. Node đóng gói các giao dịch vào một Input Block, cùng với tham chiếu đến điểm trên chuỗi, bằng chứng VRF và ký vào Block.

	Ranking Block	Endorsement Block	Input Block
Viết tắt	RB	EB	IB
Mục đích	-Đồng thuận -Sắp xếp	-Sự tồn tại của IB -Hiệu lực của IB	Thực hiện các giao dịch
Tham chiếu	-Một RB -Nhiều EB	-Nhiều IB -Không hoặc nhiều EB	Một RB
Bao gồm	Chứng chỉ xác nhận		Giao dịch
Tính thường xuyên	1 trên 15-30 giây	1 trên 5-10 giây	1 trên 0,2-2 giây

Bảng 1: Tóm tắt các loại Block khác nhau, mục đích và mối quan hệ của chúng

4.1.2 Chuyển tiếp và xác thực các Input Block

Input Block được cung cấp cho các Node ngang hàng và được chuyển tiếp đến tất cả các Node khác. Khi các Node tạo Block khác nhận được nó, chúng sẽ xác thực chữ ký, bằng chứng VRF và các giao dịch bên trong, sử dụng trạng thái sổ cái tương ứng với Ranking Block mà Input Block tham chiếu (nếu có). Lưu ý rằng các Input Block không phụ thuộc vào nhau, tất cả chúng có thể được chuyển tiếp và xác thực độc lập với nhau. Về nguyên tắc, các Input Block có thể được tạo ở tốc độ cao bằng cách điều chỉnh ngưỡng xổ số VRF. Đây là cách mà Ouroboros Leios có thể tạo ra nhiều công việc hữu ích để làm, và cũng quan trọng không kém là nó có thể phân bổ công việc tương đối đồng đều theo thời gian.

4.1.3 Tạo Endorsement Block

Các Node cũng có thể được bầu chọn (bằng một xổ số VRF có trọng số cổ phần riêng tư khác) để tạo ra một Endorsement Block. Chúng tập hợp các tham chiếu đến các Input Block hợp lệ đã thấy gần đây chưa được đưa vào các Endorsement Block khác. Đôi khi, chúng cũng có thể tham chiếu các Endorsement Block gần đây khác chưa được đưa vào Ranking Block và nơi có thể tạo chứng chỉ xác nhận cho chúng (sẽ được mô tả ngay trong Phần 4.1.5). Chúng cũng bao gồm chữ ký Block thông thường và bằng chứng VRF. Các Endorsement Block này là cơ sở để các Node khác đưa ra các báo cáo chứng thực trên các Endorsement Block để cho biết liệu tất cả các Input Block được tham chiếu bởi Endorsement Block có thực sự tồn tại và hợp lệ hay không. Các Endorsement Block được thực hiện ít thường xuyên hơn các Input Block nhưng thường xuyên hơn các Ranking Block. Lưu ý rằng sẽ ổn nếu các Endorsement Block khác nhau được tạo đồng thời bởi các Node khác nhau và tham chiếu nhiều Input Block giống nhau.

4.1.4 Chuyển tiếp Endorsement Block và tạo báo cáo chứng thực

Endorsement Block được chuyển tiếp đến tất cả các Node khác theo cách thông thường. Khi các Node tạo Block khác nhận được nó, chúng sẽ cất nó đi cho đến khi chúng có thể cần tạo một báo cáo chứng thực cho nó. Các Endorsement Block phải được kiểm tra và báo cáo trên một số Slot cố định sau khi chúng được tạo. Số Slot là một tham số giao thức được chọn để cho phép đủ thời gian để Endorsement Block được chuyển tiếp qua mạng để tiếp cận tất cả các Node sẽ thực hiện kiểm tra. Ngoài ra, còn có một xổ số VRF có trọng số cổ phần riêng tư khác, nơi các nhà sản xuất Block (Block Producer) có thể được bầu làm người báo cáo. Nếu một Node đã được chọn làm người báo cáo trong Slot mà Endorsement Block phải được kiểm tra thì nó sẽ kiểm tra xem tất cả các Input Block được tham

chiếu bởi Endorsement Block đã được nhìn thấy và xác minh chư[8]. Nếu Endorsement Block tham chiếu đến các Endorsement Block khác thì nó phải kiểm tra xem nó đã xem đủ báo cáo chứng thực cho chúng hay chưa để có thể tập hợp chứng chỉ xác nhận cho chúng (xem Phần 4.1.5 để biết chi tiết). Sau đó, Node sẽ tạo một báo cáo chứng thực đã ký và chuyển nó đến tất cả các Node khác theo cách thông thường. Báo cáo này chứa chữ ký, bằng chứng VRF thông thường và tham chiếu đến Endorsement Block được đề cập. Lưu ý rằng rất nhiều Block Producer dự kiến sẽ được bầu làm người báo cáo cùng một lúc nên sẽ có rất nhiều báo cáo chứng thực được tạo và chuyển tiếp trên mạng lưới.

4.1.5 Tạo chứng chỉ xác nhận

Các Endorsement Block và báo cáo chứng thực cho các Block đó được chuyển tiếp đến các Node tạo Block. Khi đã thu thập đủ các báo cáo chứng thực cho Endorsement Block thì có thể tạo chứng chỉ xác nhận cho Endorsement Block. Việc bầu chọn các người báo cáo được hiệu chỉnh sao cho có đủ số lượng và họ được lấy mẫu tương đối từ cổ phần để chúng ta có thể nhận được "đủ" báo cáo chứng thực. Ngưỡng "đủ" dựa trên đối số thống kê cho phép chúng ta kết luận (với mức độ tin cậy cao) rằng > 50% cổ phần xác nhận Endorsement Block nhất định, do đó xác nhận sự tồn tại và tính hợp lệ của tất cả các Input Block được tham chiếu bởi Endorsement Block (trực tiếp hoặc thông qua tham chiếu đến các Endorsement Block khác). Khi có thể thu thập được ngưỡng báo cáo chứng thực như vậy thì chúng có thể được tập hợp thành chứng chỉ xác nhận.

4.1.6 Tạo Ranking Block

Các Node có thể được bầu (bằng một xổ số VRF có trọng số cổ phần riêng tư khác) để tạo ra một Ranking Block. Chúng thu thập các tham chiếu đến các Endorsement Block gần đây chưa được đưa vào các Ranking Block trước đó, với điều kiện là Node có thể xây dựng và bao gồm chứng chỉ xác nhận cho mỗi Block. Điều này thường có nghĩa là không thể đưa vào các Endorsement Block mới nhận được vì chưa có đủ báo cáo chứng thực để tạo chứng chỉ xác nhận tương ứng. Các Endorsement Block như vậy thường sẽ được đưa vào Ranking Block tiếp theo. Các Ranking Block cũng tạo thành "xương sống" của chuỗi tổng thể, với mỗi Ranking Block tham chiếu đến Block trước đó. Chúng cũng bao gồm chữ ký Block thông thường và bằng chứng VRF.

[8]Thông số kỹ thuật là kiểm tra được thực hiện *như thể* nó được thực hiện tại Slot cụ thể, nhưng bất kỳ kiểm tra tương đương nào cũng được chấp nhận. Ví dụ: bạn có thể kiểm tra Endorsement Block sớm hơn, nhưng nếu không phải tất cả các Input Block đã đến thì việc kiểm tra phải bị hoãn lại.

Các Endorsement Block có thể tham chiếu lẫn nhau, nhưng khi chúng được đưa vào một Ranking Block, chỉ Block cuối cùng trong chuỗi mới được đưa vào chứng chỉ xác nhận. Điều này là do chứng chỉ xác nhận cho Endorsement Block tham chiếu đến các Endorsement Block trước đó hoàn toàn bao gồm các Endorsement Block trước đó.

Có giới hạn về số Endorsement Block có thể được tham chiếu trực tiếp trong Ranking Block (cùng với chứng chỉ xác nhận tương ứng của chúng). Điều này là để hạn chế chi phí xác thực Ranking Block. Nếu một Node có quyền lựa chọn về việc bao gồm các Endorsement Block nào để phù hợp với giới hạn thì Node đó nên chọn những Block "lớn nhất", nghĩa là những Block (trực tiếp hoặc gián tiếp) tham chiếu đến nhiều Input Block nhất.

4.1.7 Chuyển tiếp và xác thực các Ranking Block

Một Ranking Block mới được tạo sẽ được cung cấp cho các Node ngang hàng trực tiếp của nó thông qua giao thức đồng bộ hóa chuỗi. Các Node ngang hàng khác nhận và xác thực chuỗi mở rộng. Quy tắc áp dụng chuỗi ở đây *gần* giống như trong Praos: chuỗi hợp lệ dài nhất sẽ thắng (với các quy tắc thông thường về phá vỡ ràng buộc xác định công bằng). Mỗi Ranking Block được coi là hợp lệ nếu nó tham chiếu đến Ranking Block hợp lệ trước đó và chứa tham chiếu đến các Endorsement Block có chứng chỉ xác nhận hợp lệ tương ứng. Điều này là đủ để kết luận chuỗi hợp lệ và áp dụng nó. *Không* nhất thiết phải tải xuống hoặc xác thực tất cả các Endorsement Block và Input Block mà chúng tham chiếu. Chuỗi hợp lệ được thông qua ngay lập tức, trong khi việc xây dựng trạng thái sổ cái tương ứng sẽ được lùi lại phía sau.

4.1.8 Xây dựng trạng thái sổ cái cho chuỗi Ranking Block

Sau khi áp dụng một chuỗi mới, Node phải xây dựng trạng thái sổ cái tương ứng. Nếu cần, điều này có thể liên quan đến việc tải xuống bất kỳ Endorsement Block và đầu vào bị thiếu nào được tham chiếu bởi các Ranking Block mới trong chuỗi.

Trạng thái sổ cái phải được xây dựng *như thể* tất cả các giao dịch trong tất cả các Input Block (gián tiếp) được tham chiếu bởi mỗi Ranking Block được đặt theo thứ tự và được xác thực theo thứ tự đó (xem Phần 4.5 để biết chi tiết về thứ tự).

Tuy nhiên, có một ngoại lệ quan trọng: bất kỳ giao dịch nào được phát hiện là xung đột đều được coi như *không có trong sổ cái*. Điều này có nghĩa là trong một tập hợp các giao dịch xung đột, giao dịch đầu tiên (trong thứ tự giao dịch) được chấp nhận và các giao dịch còn lại bị bỏ qua. Lưu ý rằng các giao dịch bị bỏ qua vẫn ở dạng vật lý trong các Input Block ban đầu của chúng nên vẫn sẽ chiếm dung lượng, nhưng về mặt Logic thì chúng không phải là một phần của sổ cái.

Lưu ý rằng điều này xác định kết quả cho lớp sổ cái chứ không xác định cách đạt được kết quả đó. Điều quan trọng đối với hiệu suất là việc xây dựng trạng thái

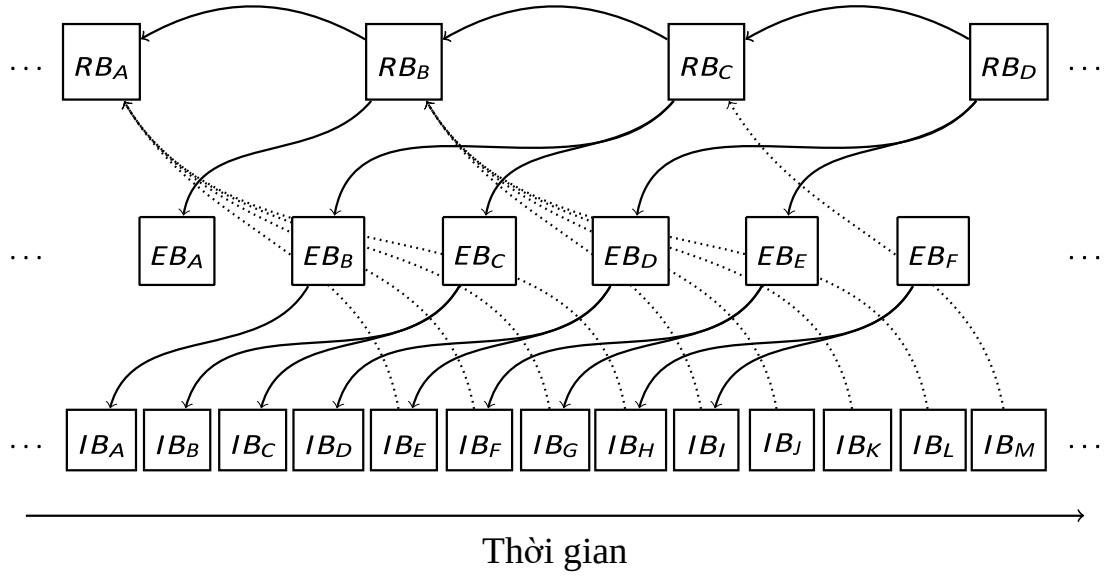
sổ cái Ranking Block liên quan đến tính toán ít hơn so với việc xác thực đầy đủ tất cả các giao dịch theo thứ tự. Xem Phần 4.6 để biết thêm chi tiết.

4.2 Cấu trúc Blockchain

Cấu trúc Blockchain Ouroboros Leios bao gồm ba loại khối khác nhau, cộng với các báo cáo chứng thực:

1. Ranking Block (RB)
2. Endorsement Block (EB)
3. Endorsement Report (ER)
4. Input Block (IB)

Xem Hình 5 để biết minh họa.



Hình 5: Sự phụ thuộc dữ liệu giữa các Block trong Ouroboros Leios: mỗi Ranking Block (RB) tham chiếu đến Ranking Block trước đó và các Endorsement Block trước đó (EB), chúng tham chiếu đến các Input Block (IB) trước đó. Các Input Block cũng tham chiếu các Ranking Block trước đó.

4.2.1 Ranking Block

Mục đích của các Ranking Block là để đạt được sự đồng thuận và thu xếp tổng thể. Mỗi Ranking Block tham chiếu đến Ranking Block trước đó và tham chiếu không hoặc nhiều Endorsement Block (và bao gồm các chứng chỉ xác nhận tương ứng của chúng). Do đó, các Ranking Block tạo thành một Blockchain tuyến tính truyền thống, nhưng thay vì chứa các giao dịch, chúng chứa tham chiếu đến các

Endorsement Block. Thật vậy, các Ranking Block tạo thành một chuỗi Ouroboros Praos, cho phép các kết quả phân tích và bảo mật từ Ouroboros Praos và Genesis được chuyển sang.

Như trong Praos, các Ranking Block được tạo bằng cách sử dụng lịch trình lãnh đạo riêng tư dựa trên VRF và được tính theo cổ phần. Điều này có nghĩa là các Ranking Block có mẫu đến ngẫu nhiên tuân theo phân phối Poisson, trong đó thời gian trung bình giữa các Block có thể được điều chỉnh khi cần thiết. Đối với các triển khai toàn cầu, thời gian trung bình giữa các Block dự kiến sẽ tương tự như 20 giây được sử dụng trên mạng chính Cardano hiện tại, nhưng các giá trị trong khoảng 15-30 giây có thể hợp lý.

4.2.2 Endorsement Block

Mục đích của các Endorsement Block là giúp thống nhất về sự tồn tại và hiệu lực của các Input Block. Chúng tập hợp một nhóm gồm nhiều Input Block để cho phép báo cáo về chúng dưới dạng một nhóm. Điều này khấu hao chi phí xây dựng và truyền báo cáo qua nhiều Input Block.

Các Endorsement Block được tạo bằng cách sử dụng lịch trình lãnh đạo riêng tư dựa trên VRF và được tính theo tỷ lệ cổ phần. Tần suất tạo dự kiến sẽ được điều chỉnh sao cho có khoảng 2-4 Endorsement Block trên mỗi Ranking Block.

4.2.3 Báo cáo chứng thực và chứng chỉ xác nhận

Tổng hợp lại, mục đích của các báo cáo chứng thực là để chứng minh sự tồn tại và tính hợp lệ của gói các Input Block được tham chiếu bởi Endorsement Block. Ở dạng tổng hợp của chúng, với đủ chúng để đại diện theo thống kê > 50% cổ phần, chúng tôi gọi chúng là chứng chỉ xác nhận.

Chứng chỉ xác nhận là thứ giúp có thể áp dụng một chuỗi các Ranking Block ngay cả khi chưa nhìn thấy tất cả các Input Block và Endorsement Block được tham chiếu: chúng tôi biết rằng sẽ có thể tìm thấy các Block đó và chúng sẽ hợp lệ. Chúng tôi biết rằng trên cơ sở chứng chỉ xác nhận chứng minh (với mức độ tin cậy cao) rằng các Block Producer đại diện cho phần lớn cổ phần đã xem các Input Block và kiểm tra xem chúng có hợp lệ hay không. Điều này cũng tạo cơ hội cho những người tiêu dùng Blockchain khác bỏ qua việc xác thực các Input Block và dựa vào các chứng chỉ xác nhận để đảm bảo tính hợp lệ.

4.2.4 Input Block

Mục đích của các Input Block là mang trọng tải Blockchain: các giao dịch. Mỗi Input Block chứa một chuỗi các giao dịch. Nó cũng tham chiếu đến một Ranking Block gần đây. Tham chiếu này đến một Ranking Block là để làm rõ trạng thái sổ cái nào được sử dụng để xác thực các giao dịch trong một Input Block. Các giao dịch trong một Input Block có thể phụ thuộc lẫn nhau.

Các Input Block cũng được tạo bằng cách sử dụng lịch trình lãnh đạo riêng tư bằng VRF. Chúng dự định được tạo ra với tốc độ cao. Thông lượng của chuỗi Leios chủ yếu được xác định bởi tốc độ tạo và kích thước tối đa của Input Block. Tốc độ tạo hợp lý có thể là từ một Input Block cứ sau vài giây, lên đến vài Input Block mỗi giây.

4.3 Chi tiết thuật toán khác

Ngoài phần tổng quan trong Phần 4.1, còn có một số chi tiết khác về thuật toán.

4.3.1 Mempool Sharding

Có khả năng có một số lượng đáng kể tính đồng thời có sẵn giữa nhiều Input Block "đang hoạt động", mang lại khả năng có băng thông dữ liệu cao. Tuy nhiên, tất cả điều này có thể bị lãng phí nếu hầu hết các Block đồng thời chứa hầu hết các giao dịch giống nhau. Đây là một mối nguy hiểm thực sự. Người ta mong đợi rằng hai Node tạo Input Block cùng một lúc hoặc cách nhau vài giây sẽ có nội dung Mempool rất giống nhau. Do đó, một cách đơn giản để chọn các giao dịch từ Mempool sẽ tạo ra các Block với một mức độ chồng chéo cao.

Có nhiều giải pháp tiềm năng cho vấn đề này, việc tạo mẫu và mô phỏng thêm sẽ được yêu cầu để chọn ra thiết kế tốt nhất. Thiết kế ứng viên tốt nhất hiện tại như sau. Tác giả giao dịch gán cho mỗi giao dịch một "màu" từ một tập hợp. Điều này được thể hiện dưới dạng một số từ một phạm vi giới hạn. Tác giả giao dịch nên chỉ định màu này một cách ngẫu nhiên, trừ khi họ muốn gửi liên tiếp nhiều giao dịch phụ thuộc, trong trường hợp đó, họ nên xác định các giao dịch tiếp theo cùng màu. Nhiều tác giả giao dịch hợp tác tạo giao dịch phụ thuộc cũng có thể sắp xếp việc này.

Khi một Node đến để tạo một Input Block, nó sẽ tạo ra một chuỗi màu ngẫu nhiên, sau đó với mỗi màu lần lượt, nó sẽ chọn tất cả các giao dịch từ Mempool có màu đó. Nó tiếp tục chọn các giao dịch theo cách này cho đến khi Input Block đầy. Bằng cách này, các Node khác nhau thường sẽ chọn các giao dịch khác nhau, nhưng các giao dịch phụ thuộc có thể được giữ cùng nhau.

Trong quá trình triển khai Cardano hiện tại, Mempool có kích thước tỷ lệ thuận với lượng dữ liệu đang được xử lý: gấp đôi kích thước khối. Trong Ouroboros Leios, với tất cả các Input Block đồng thời, sẽ có nhiều dữ liệu đang được xử lý hơn và kích thước Mempool sẽ cần phải được xác định theo tỷ lệ.

4.3.2 Thời gian tối thiểu và tối đa để đưa vào

Có một khoảng thời gian tối thiểu giữa khi một Input Block được tạo và khi nó được phép tham chiếu trong một Endorsement Block. Tương tự như vậy, có một khoảng thời gian tối đa sau đó Input Block không còn đủ điều kiện để được tham

chiếu trong Endorsement Block. Tương tự như vậy, có thời gian tối thiểu và tối đa cho các Endorsement Block được tham chiếu trong các Ranking Block.

Những thời điểm này được chọn để có một khoảng thời gian hợp lý trong đó các Block đủ điều kiện để đưa vào. Kết quả là các Input Block không được tham chiếu bởi Endorsement Block khả thi đầu tiên có thể được tham chiếu bởi các Endorsement Block tiếp theo hoặc tiếp theo. Điều này cung cấp một mức độ bảo vệ hợp lý khỏi sự kiểm duyệt, vì nhiều nhà sản xuất Endorsement Block (được lấy mẫu khá tốt từ phân phối cổ phần) sẽ có cơ hội bao gồm từng Input Block trước khi nó hết hạn khi đạt đến giới hạn thời gian tối đa. Điều này áp dụng tương tự cho các Endorsement Block được bao gồm trong các Ranking Block.

Giới hạn tối đa là khả năng bảo vệ chống lại các Node tạo Block độc hại có quyền tạo nhiều Block nhưng điều đó làm chậm quá trình thực hiện và sau đó giải phóng tất cả chúng cùng một lúc. Giới hạn tối đa giới hạn quy mô của một cuộc tấn công tràn ngập như vậy. Phần khác của phòng thủ là ưu tiên mạng được thảo luận ở nơi khác.

Giới hạn tối thiểu phục vụ hai mục đích: một là phần của bảo vệ ký hai lần (Double Signing) như sẽ được thảo luận ngay sau đây và thứ hai là nó cung cấp đủ thời gian để Block được chuyển tiếp qua mạng. Cung cấp đủ thời gian có nghĩa là không có động cơ vô ích nào để các Block Producer cạnh tranh để nhanh hơn một chút và đặc biệt là không có động cơ nào để các Block Producer tập trung lại gần nhau về mặt địa lý, điều này sẽ chống lại khía cạnh quan trọng này của quá trình phi tập trung.

4.3.3 Bảo vệ ký hai lần

Thuật toán bao gồm các biện pháp đặc biệt để phát hiện và xử lý tình huống trong đó một Node độc hại hoặc được định cấu hình kém ký nhiều Input Block hoặc Endorsement Block khác nhau trong cùng một Slot. Lưu ý rằng điều này có thể vô tình xảy ra khi một SPO thiết lập hệ thống chuyển đổi dự phòng "hoạt động/hoạt động" hoặc "hoạt động/thụ động", nếu có sự nhầm lẫn dẫn đến nhiều Node tin rằng đó là Node hoạt động và do đó ký các Block. Sự bảo vệ này chỉ cần thiết cho các Input Block và Endorsement Block, không phải các Ranking Block, đó là lý do tại sao điều này không cần thiết trong Ouroboros Praos[9]. Các Node theo dõi các Input Block và Endorsement Block gần đây như một phần của quy trình thông thường. Nếu họ nhận được Block thứ hai được ký bởi cùng một Block

[9]Lý do khiến các Ranking Block, hay thực sự là các Block Praos, không cần bảo vệ ký hai lần hoạt động là vì chúng là một phần của chuỗi và cách các chuỗi được chọn một cách tự nhiên cung cấp khả năng bảo vệ DoS. Các Node chỉ tải xuống và sử dụng các chuỗi tốt hơn chuỗi hiện tại của chính chúng. Hai chuỗi có độ dài bằng nhau kết thúc bằng hai Block khác nhau được ký bởi cùng một Node ngang hàng không tốt hơn nhau trong thứ tự chuỗi, vì vậy các Node sẽ chọn chuỗi đầu tiên mà chúng nhìn thấy và sau đó không chọn chuỗi thứ hai.

Producer cho cùng một số Slot thì Block thứ hai này sẽ được thêm vào bộ theo dõi và Block (hoặc ít nhất là tiêu đề) được chuyển tiếp tiếp theo để hầu hết các Node khác cũng sẽ thấy Block trùng lặp. Bất kỳ Block trùng lặp tiếp theo nào từ cùng một người ký trong cùng một Slot sẽ không được chuyển tiếp để tránh bị từ chối dịch vụ. Điều này đảm bảo rằng hầu hết các Node tạo Block sẽ thấy rằng có một bản sao cho một người ký cụ thể (mặc dù chúng có thể có các Block khác nhau làm bằng chứng về điều này).

Giờ đây, khi một Node tạo Endorsement Block, quy tắc sẽ phức tạp hơn so với việc chỉ chọn tất cả các Input Block hợp lệ có sẵn. Tất nhiên, không được bao gồm các Input Block nằm trong bộ theo dõi ký hai lần. Hơn nữa, phải có đủ thời gian để phát hiện việc ký hai lần nên quy tắc là có một số Slot cố định sau khi Input Block được tạo, trước đó nó không được phép đưa vào Endorsement Block. Số Slot là một tham số giao thức được chọn để cho phép đủ thời gian cho bất kỳ Input Block trùng lặp nào (tiêu đề) được chuyển tiếp qua mạng. Các quy tắc bao gồm các Endorsement Block vào các Ranking Block hoạt động tương tự nhau.

4.3.4 Ưu tiên tài nguyên mạng

Một Block Producer có cổ phần đáng kể có thể cố gắng tiến hành một cuộc tấn công từ chối dịch vụ bằng cách giữ lại một số lượng lớn các Block hoặc báo cáo chứng thực và sau đó phát hành tất cả chúng cùng một lúc. Đây là mối quan tâm đặc biệt đối với các Input Block và báo cáo chứng thực vì chúng được tạo với tần suất tương đối cao. Điều này sẽ dẫn đến tình huống có thể có nhiều Block hoặc báo cáo có sẵn để tải xuống hơn là tất cả có thể được tải xuống cùng một lúc trong thời gian ngắn. Nó cũng cung cấp cơ hội để chọn những gì cần tải xuống hoặc theo thứ tự nào. Hầu hết các Block và báo cáo hợp pháp sẽ có số Slot gần đây, trong khi những Block và báo cáo đó trong vùng ngập sẽ có hầu hết các số Slot cũ hơn. Do đó, các Node phải luôn chọn các Block và báo cáo gần đây nếu có và chỉ chọn những Block cũ hơn nếu chúng là những Block duy nhất có sẵn. Ngưỡng cho "gần đây" ở đây là một tham số giao thức dựa trên giới hạn trên hợp lý về thời gian điển hình cần thiết để các Block và báo cáo được chuyển tiếp qua mạng.

Ngoài ra, khi có các lựa chọn tải xuống nội dung nào trong băng thông hạn chế, các Ranking Block nên được chọn trước các loại Block và báo cáo khác.

4.3.5 Trạng thái sổ cái tụt lùi so với các Ranking Block

Như đã lưu ý ngắn gọn trong phần tổng quan, có thể áp dụng một chuỗi mặc dù không phải tất cả các Input Block và Endorsement Block được tham chiếu đều chưa được tải xuống và xác minh. Điều này có một vài hậu quả cần được giải thích. Điều đó có nghĩa là trạng thái của một Node có một chuỗi hiện tại nhưng nói chung nó sẽ chỉ có trạng thái sổ cái cho một Ranking Block gần đây và không nhất thiết là Ranking Block mới nhất. Điều đó cũng có nghĩa là Mempool sẽ được

xác thực lại theo Ranking Block cuối cùng mà chúng ta có trạng thái sổ cái thay vì Ranking Block mới nhất.

Một Node ở trạng thái "không cập nhật đầy đủ" này nên ưu tiên tải xuống tất cả các Input Block và Endorsement Block còn thiếu để cố gắng bắt kịp, bởi vì có một số hoạt động mà nó không thể tham gia khi không cập nhật. Các Node cần trạng thái sổ cái cho một Ranking Block để có thể xác thực các Input Block tham chiếu đến Ranking Block đó. Do đó, các Node không được cập nhật đầy đủ không thể tạo báo cáo chứng thực cho các Input Block mới tham chiếu Ranking Block gần đây hơn. Tương tự, một Node như vậy được chọn để tạo Endorsement Block có thể phải chuyển qua nhiều Input Block nếu nó không thể xác thực chúng. Và tất nhiên, các ứng dụng Blockchain muốn lấy nội dung của các Block, nghĩa là có quyền truy cập vào các Input Block.

4.3.6 Chế độ bi quan

Trong những điều kiện khắc nghiệt khi sự tham gia giảm xuống và các Block Producer đang hoạt động còn lại chiếm một phần cổ phần quá nhỏ, có thể có quá ít Block Producer tạo báo cáo chứng thực để có thể tạo chứng chỉ xác nhận hợp lệ nên tiến độ sẽ bị đình trệ. Để giảm thiểu một cách nhẹ nhàng trong tình huống này và duy trì tính sống động, Ouroboros Leios chuyển sang hành xử giống như Ouroboros Praos, giúp duy trì tính sống động ngay cả khi cổ phần tham gia thấp.

Cách thức hoạt động của điều này là khi một Node được chọn để tạo Ranking Block, nó sẽ đánh giá một vị từ trên chuỗi gần đây, sau đó xác định xem giao thức nên ở chế độ "lạc quan" hay "bi quan". Vị từ dựa trên số lượng Input Block trên chuỗi gần đây so với phạm vi dự kiến. Ở chế độ lạc quan, mọi thứ diễn ra như mô tả trước đây. Ở chế độ bi quan, Ranking Block được tạo trực tiếp chứa các giao dịch mà không có bất kỳ Endorsement Block nào. Điều này làm cho Block giống như Praos Block.

4.3.7 Thời gian Slot của một giao dịch

Các giao dịch Cardano có khoảng thời gian hiệu lực được biểu thị bằng các thời gian Slot. Điều này có nghĩa là các giao dịch chỉ có thể được đưa vào chuỗi sau một thời gian Slot nhất định và trước một thời gian Slot khác. Khoảng thời gian hiệu lực này là cơ sở cho khái niệm thời gian trong tập lệnh Plutus.

Trong Ouroboros Praos, việc giải thích thời gian Slot là Slot của Block mà giao dịch được đưa vào. Trong Ouroboros Leios, tình hình ít rõ ràng hơn và có nhiều lựa chọn với những sự đánh đổi khác nhau. Phân tích sâu hơn là cần thiết để quyết định lựa chọn tốt nhất.

Một tùy chọn là tuyên bố rằng việc giải thích thời gian Slot của giao dịch là Slot của *Input Block* mà giao dịch được đưa vào. Điều này có nghĩa là một giao dịch có thể được đưa vào Input Block ngay trước thời hạn của giao dịch, nhưng

sau đó phải mất thêm một khoảng thời gian trước khi Input Block được tham chiếu trong Endorsement Block được tham chiếu trong Ranking Block. Điều này theo một nghĩa nào đó là hợp lý: giao dịch đã được gửi và chuyển đến Block Producer đúng hạn, chỉ đơn giản là quá trình xây dựng chuỗi sẽ mất một thời gian sau đó để hoàn thành việc tập hợp mọi thứ. Một lợi thế của điều này là nó rất đơn giản để thực hiện: kiểm tra khoảng thời gian có thể được thực hiện cùng với tất cả các kiểm tra khác khi các giao dịch trong một Input Block được xác thực. Nhược điểm của lựa chọn này là để chắc chắn rằng một giao dịch *chưa* đến trước thời hạn sẽ liên quan đến việc chờ thời gian tối đa mà các Input Block được phép đưa vào các Ranking Block. Mặt khác, về nguyên tắc, một giao dịch có thể đã được đưa vào một Input Block, nhưng Input Block đó đã bị một số Block Producer trì hoãn hoặc bỏ qua và cuối cùng được đưa vào rất muộn. Mức độ hợp lý của điều này phụ thuộc vào khoảng thời gian các Input Block được phép tồn tại trước khi được đưa vào Ranking Block, do đó đây là một tham số giao thức cần phân tích thêm để hiệu chỉnh chính xác. Hơn nữa, tùy chọn này có nghĩa là các giao dịch xuất hiện trong chuỗi cuối cùng không nhất thiết phải xuất hiện theo thứ tự thời gian Slot: một lần nữa vì các Input Block có thể bị trì hoãn và đưa vào muộn.

Một tùy chọn khác là tuyên bố rằng việc giải thích thời gian Slot là Slot của Ranking Block mà giao dịch được đưa vào. Điều này sẽ có lợi thế là dễ dàng biết khi nào một giao dịch chưa đến trước thời hạn, vì điều đó chỉ liên quan đến việc chờ đợi Ranking Block đầu tiên sau thời hạn. Điều đó cũng có nghĩa là các giao dịch sẽ xuất hiện theo thứ tự thời gian Slot trong sổ cái cuối cùng, vì mọi giao dịch được tham chiếu gián tiếp bởi một Ranking Block sẽ được cung cấp thời gian Slot của Ranking Block đó. Cũng có một số nhược điểm. Có nguy cơ các giao dịch được đưa vào các Input Block mà sau đó phải bỏ qua vì chúng nằm ngoài khoảng thời gian hiệu lực của chúng và trừ khi được quản lý cẩn thận, điều này có thể tạo ra cơ hội từ chối dịch vụ. Nó làm phức tạp việc kiểm tra sổ cái vì nó có nghĩa là một số kiểm tra phải được phân chia và thực hiện ở các giai đoạn khác nhau trong thuật toán. Điều đó cũng có nghĩa là có nhiều sự không chắc chắn hơn khi gửi giao dịch trước thời hạn và đưa chúng vào một cách đáng tin cậy: sự chậm trễ trong quá trình xử lý Endorsement Block hoặc Ranking Block có thể có nghĩa là giao dịch bỏ lỡ thời hạn và điều này nằm ngoài tầm kiểm soát của tác giả giao dịch.

4.4 Bình luận thuật toán

4.4.1 Chuyển đổi nhánh nhanh

Một trong những động lực cho chương trình chứng chỉ xác nhận trong Ouroboros Leios là để giải quyết sự không phù hợp phức tạp giữa lý thuyết và thực tế trong thiết kế Ouroboros Praos cũ mà thậm chí còn trở nên tồi tệ hơn trong Ouroboros Leios. Lý thuyết và phân tích bảo mật cho Ouroboros Praos dựa trên giả định rằng các chuỗi "phân tán" trong thời gian Slot Δ (5 giây trên Cardano). Đây là điều mà

việc triển khai thực tế có thể thực hiện đối với các Block đơn lẻ và đối với các nhánh ngắn của một vài Block. Tuy nhiên, rõ ràng là thời gian để chuyển đổi nhánh ít nhất là tỷ lệ thuận với độ dài của nhánh và ở một số độ dài sẽ dài hơn thời gian Slot Δ. Vấn đề này không phải là một vấn đề nghiêm trọng trong Cardano ngày nay với Praos, một phần vì các Block không quá lớn. Tuy nhiên, nó sẽ trở thành một vấn đề lớn ở Leios vì nó được thiết kế để có nhiều dữ liệu và tập lệnh hơn trên chuỗi. Do đó, chi phí xác thực tất cả dữ liệu được tham chiếu bởi mỗi Ranking Block sẽ cao hơn nhiều. Điều này thực sự sẽ vi phạm giả định của Praos rằng các chuỗi có thể được khuếch tán trong thời gian Δ, bao gồm cả việc chuyển đổi nhánh. Nó có thể bị vi phạm đối với các nhánh khá ngắn.

Các chứng chỉ xác nhận cung cấp một cách để chuyển đổi các nhánh trên chuỗi Ranking Block với tương đối ít dữ liệu cần tải xuống và CPU xác nhận tương đối ít hoạt động. Điều này khôi phục giả định về thời gian khuếch tán Δ.

Tải trọng Ranking Block chỉ là tham chiếu đến các Endorsement Block và chứng chỉ xác nhận. Những chứng chỉ này không nhỏ nhưng cũng không quá lớn. Các Ranking Block có thể được chấp nhận sau khi xác minh các chứng chỉ xác nhận mà không cần phải tải xuống hoặc xác minh tất cả các Endorsement Block hoặc Input Block.

4.4.2 Chuỗi nhẹ hơn

Chứng chỉ xác nhận cũng kích hoạt tính năng mới cho các Node không tạo Block, chẳng hạn như Node Relay và Node người dùng cuối. Nó cho phép họ bắt kịp và theo dõi chuỗi *mà không* cần phải thực hiện hợp đồng thông minh và xác minh chữ ký giao dịch. Họ có thể xây dựng lại trạng thái sổ cái cho chuỗi mà họ đang theo dõi bằng cách áp dụng các Block, nhưng khi làm như vậy, họ có thể bỏ qua việc xác minh việc thực thi các tập lệnh và xác minh chữ ký giao dịch. Họ có thể bỏ qua các bước kiểm tra này vì chứng chỉ xác nhận cho biết rằng các Node đại diện cho phần lớn cổ phần đã xác minh các tập lệnh thực thi như mong đợi.

Điều đáng chú ý là trong Cardano, việc thực thi tập lệnh không ảnh hưởng đến "kết quả" của giao dịch vì điều đó đã được cung cấp trong kết quả đầu ra của giao dịch, nó chỉ ảnh hưởng đến việc liệu giao dịch có hợp lệ hay không. Nếu giao dịch được biết là hợp lệ thì trạng thái sổ cái kết quả có thể được tính toán mà không thực thi các tập lệnh. Điều này cũng đúng với chữ ký trên các giao dịch.

Kết quả của việc này là có thể thực hiện theo chuỗi với ít tài nguyên CPU hơn so với cách khác. Tuy nhiên, dữ liệu chuỗi vẫn cần thiết. Mặc dù về nguyên tắc giao dịch nhân chứng không cần phải được tải xuống.

4.4.3 Relay nhẹ hơn

Trong Ouroboros Praos, chi phí tài nguyên để vận hành Relay về cơ bản giống với chi phí tài nguyên để vận hành một Node tạo Block. Node tạo Block thực hiện một số công việc bổ sung, nhưng nó tương đối nhỏ.

Trong Ouroboros Leios, về nguyên tắc, có thể có các Relay hoạt động ít hơn so với các Node tạo Block. Đặc biệt, các Relay không cần xác minh nội dung của các Input Block mà chúng chuyển tiếp. Đối với các Input Block, Endorsement Block và báo cáo chứng thực, số của chúng được giới hạn với điều kiện VRF và chữ ký được kiểm tra, đồng thời thực hiện kiểm tra ký hai lần. Điều này là đủ để ngăn chặn DoS.

Relay vẫn phải đi theo chuỗi, nhưng chúng có thể sử dụng kỹ thuật được mô tả ở trên để làm như vậy với chi phí tương đối rẻ. Relay sẽ vẫn cần bộ nhớ và dung lượng ổ đĩa để giữ trạng thái sổ cái và băng thông mạng để phục vụ tất cả các Node ngang hàng ngược dòng và xuôi dòng của chúng, nhưng không cần nhiều sức mạnh xử lý CPU như các Node tạo Block.

4.4.4 Lịch trình dựa trên VRF riêng tư

Quyền tạo các đối tượng khác nhau (Input Block, Endorsement Block, báo cáo chứng thực và Ranking Block) đều dựa trên VRF (chức năng ngẫu nhiên có thể kiểm chứng). Điều này giống như lịch trình lãnh đạo riêng tư để tạo các Block thông thường trong Ouroboros Praos. Sự khác biệt duy nhất là hiện có bốn lịch trình như vậy, và điều quan trọng là mỗi lịch trình đều độc lập với lịch trình khác. Mỗi Node tạo Block có thể tạo bất kỳ đối tượng nào trong số bốn đối tượng khác nhau và tham gia vào lịch trình tương ứng.

Mỗi lịch trình hoạt động theo cùng một cách, dựa trên VRF và được cân bằng bởi cổ phần được ủy quyền của Node tạo Block. Lịch trình cho bốn loại khác nhau đều có các ngưỡng khác nhau được điều chỉnh để tạo ra các nhà lãnh đạo ở các tỷ lệ (trung bình) khác nhau. Ví dụ: các Input Block được tạo rất thường xuyên, trong khi các Ranking Block sẽ được tạo với tốc độ tương tự như các Block Praos được tạo hiện tại.

4.4.5 Phương sai thấp hơn trong phần thưởng của Block Producer

Ouroboros Leios liên quan đến nhiều đối tượng trên chuỗi hơn trên mỗi Epoch so với Ouroboros Praos nên có nhiều cơ hội tham gia hơn. Phần thưởng cho việc tham gia sẽ được phân bổ tương tự cho các đối tượng khác nhau trên chuỗi.

Trong quá trình triển khai Cardano hiện tại, trung bình có 21.600 Block trên mỗi Epoch và nhiều Node tạo Block nằm trên ranh giới tạo ra 0 hoặc 1 Block trên mỗi Epoch. Mức độ chi tiết này tạo ra sự khác biệt lớn trong phần thưởng của mỗi Epoch. Với Ouroboros Leios, chúng tôi mong đợi số lượng Ranking Block trên

mỗi Epoch tương tự như hiện tại, nhưng có thể có 10x–20x hoặc nhiều Input Block hơn.

Phần thưởng nên được phân bổ như thế nào để tốt nhất giữa các đối tượng khác nhau vẫn chưa được giải quyết, nhưng rõ ràng là có nhiều đối tượng đóng góp vào phần thưởng hơn. Điều này sẽ phần nào làm dịu đi sự khác biệt giữa các Epoch đối với các Block Producer có cổ phần thấp, mà không nhất thiết ảnh hưởng đến phần thưởng (trung bình) dự kiến.

4.4.6 "Cuộc chiến" chiều cao và Slot

Trong Ouroboros Praos, cái gọi là "cuộc chiến chiều cao" (Height Battle) xảy ra bất cứ khi nào các Block được tạo đồng thời, nghĩa là không Block Producer nào nhìn thấy Block của Node kia trước khi tạo Block của riêng họ. Điều này dẫn đến việc các Block Producer thua cuộc vì chỉ một trong số các Block có thể kết thúc chuỗi cuối cùng.

Tình huống này trông khá khác trong Ouroboros Leios. Vẫn có thể có những cuộc chiến chiều cao để giành các Ranking Block, nhưng không phải đối với các Input Block và Endorsement Block, cũng như đối với các báo cáo chứng thực. Tất nhiên, các Input Block được *tạo* đồng thời. Tất cả chúng có thể được tập hợp lại và sau đó được đưa vào chuỗi xếp hạng. Điều tương tự cũng áp dụng cho các Endorsement Block.

Tình huống đối với các Ranking Block là trong khi các cuộc chiến vẫn sẽ xảy ra, chúng không nên diễn ra quá thường xuyên. Trong Praos, cách duy nhất để tăng quy mô thông lượng - các Block lớn hơn - cũng khiến các cuộc chiến chiều cao diễn ra thường xuyên hơn. Ngược lại ở Leios, kích thước của các Ranking Block được cố định ngay cả khi thông lượng được tăng lên nên thời gian thông thường để chuyển tiếp một Ranking Block sẽ cố định và tương đối thấp. Điều này sẽ giữ cố định tần suất của các cuộc chiến chiều cao Ranking Block.

4.4.7 Khuyến khích tối thiểu để được gần hơn/nhanh hơn

Phi tập trung địa lý là một thành phần quan trọng của phi tập trung mạng. Thiết kế Cardano hiện tại cố gắng tránh các khuyến khích đối với việc tập trung hóa địa lý. Điều này nghĩa là cố gắng tránh thưởng cho các Block Producer vì họ nhanh hơn, vì cách chính để nhanh hơn là ở gần hơn về mặt địa lý. Điều này là do độ trễ của mạng là một thành phần chính của thời gian cần thiết để các Block được chuyển tiếp trên mạng. Đây là một tính năng được bảo tồn trong Ouroboros Leios.

Đối với các Ranking Block, các mối quan hệ bị phá vỡ một cách xác định bằng cách sử dụng VRF công bằng (ngẫu nhiên thống nhất). Đối với các Input Block và Endorsement Block, có một khoảng thời gian tối thiểu trước khi các đối tượng này được phép tham chiếu và thời gian này được đặt để có đủ thời gian chuyển

tiếp các Block trên mạng lưới. Tương tự, các báo cáo chứng thực không được sử dụng trong Ranking Block cho đến một thời gian sau.

Hiệu quả là có động cơ để không quá chậm, nhưng không có động cơ sử dụng phần cứng có thông số kỹ thuật đặc biệt cao hoặc tiến gần hơn về mặt vật lý với các Block Producer khác. Thời gian "đủ" để chuyển tiếp các Block qua mạng sẽ cần được đặt dựa trên các yêu cầu phần cứng tối thiểu, cũng như độ trễ mạng đã biết. Đây không phải là một ràng buộc chặt chẽ.

4.5 Tuần tự hóa các giao dịch trong sổ cái

Có một thứ tự giao dịch tổng thể được xác định rõ ràng trong Blockchain Leios. Như đã thảo luận trong Phần 3.2, điều này rất quan trọng để có thể cung cấp cách giải thích đơn giản về sổ cái.

Các Ranking Block cung cấp thứ tự tổng thể. Mỗi Ranking Block (gián tiếp thông qua các Endorsement Block) tham chiếu đến một tập hợp các Input Block. Các Input Block được tham chiếu (gián tiếp) từ một Ranking Block được sắp xếp theo số Slot của chúng. Do đó, lưu ý rằng không quan trọng Endorsement Block nào được tham chiếu đến Input Block nào (hoặc nếu một số Block đã tham chiếu): tất cả các Input Block được tham chiếu bởi tất cả các Endorsement Block trong một Ranking Block trước tiên được coi là một tập hợp, sau đó được sắp xếp theo số Slot của các Input Block. Thường sẽ có các ràng buộc do nhiều Input Block được tạo trong cùng một Slot và các ràng buộc này được giải quyết bằng VRF[10]. Trong trường hợp rất khó xảy ra[11] nếu còn bất kỳ ràng buộc nào, đơn đặt hàng là lựa chọn tự do của tác giả Ranking Block hoặc Endorsement Block.

Lưu ý rằng sơ đồ này không đảm bảo rằng các Input Block sẽ xuất hiện trong sổ cái theo thứ tự số Slot của chúng, vì Input Block được tạo trước đó có thể kết thúc trong Ranking Block sau.

4.6 Yêu cầu đối với lớp sổ cái

Ouroboros Leios áp đặt các ràng buộc mới trên lớp sổ cái. Đây là hệ quả trực tiếp của tính đồng thời mới trong cấu trúc Blockchain. Nhớ lại từ Hình 5, cấu trúc đồng thời có cả "Fan out" và "Fan in" trong phần phụ thuộc dữ liệu giữa các Block. Nó có "Fan out" theo nghĩa là nhiều Input Block phụ thuộc vào Ranking Block trước đó. Điều này rất dễ xử lý bằng cách sử dụng chức năng sổ cái hiện có: mỗi Input Block được xác thực độc lập ở trạng thái sổ cái ban đầu chung. Nó có "Fan in" theo nghĩa là mỗi Ranking Block phụ thuộc (gián tiếp) vào nhiều Input Block. Đây là nơi cần có một hoạt động sổ cái mới: để tập hợp trạng thái sổ cái của một

[10]Lưu ý rằng đây là một VRF độc lập với VRF lãnh đạo Input Block nên việc phá vỡ liên kết là ngẫu nhiên thống nhất.

[11]VRF được sử dụng trong Cardano có đầu ra 256 Bit nên khả năng xảy ra va chạm là rất nhỏ.

Ranking Block dựa trên chuỗi các Input Block được xác thực mà Ranking Block (gián tiếp) tham chiếu.

Theo trực giác, hoạt động cần làm là bắt đầu với trạng thái sổ cái của Ranking Block trước đó, chạy qua tất cả các giao dịch trong Ranking Block theo thứ tự và thực hiện một số chứ không phải tất cả các kiểm tra sổ cái thông thường. Cụ thể, không cần thiết phải kiểm tra lại những thứ không phụ thuộc vào trạng thái sổ cái, do đó không thể làm mất hiệu lực do thay đổi trạng thái sổ cái. Ví dụ đơn giản, tính hợp lệ của chữ ký giao dịch không thay đổi nếu trạng thái sổ cái thay đổi. Bất kỳ giao dịch nào được phát hiện là không hợp lệ dựa trên các kiểm tra này đều được hiểu là các giao dịch đó không có trong sổ cái. Điều này sẽ xảy ra bất cứ khi nào các giao dịch từ các Input Block độc lập xung đột với nhau: ví dụ, nếu chúng chi tiêu cùng một đầu vào hoặc cả hai đều rút tiền từ cùng một tài khoản phần thưởng. Giải pháp là giao dịch đầu tiên thắng và những giao dịch tiếp theo bị bỏ qua. Những cái sau vẫn chiếm không gian trong các Input Block, nhưng chúng không được coi là một phần của sổ cái cuối cùng.

Điều rất quan trọng là hoạt động "tập hợp lại" này không quá tốn kém: hầu hết công việc xác thực cần được thực hiện khi các Input Block được xác thực độc lập chứ không phải khi chúng được tập hợp lại thành một Ranking Block. Đặc biệt, điều quan trọng là các tập lệnh không cần phải được thực thi lại trong quá trình tập hợp lại. Theo trực giác, chúng tôi hy vọng việc bỏ qua thực thi tập lệnh là hợp lệ vì sổ cái đã được thiết kế rất cẩn thận để đảm bảo rằng việc thực thi tập lệnh là xác định. Điều này có nghĩa là khi một giao dịch được kiểm tra ở trạng thái sổ cái khác thì giao dịch đó không hợp lệ (ví dụ: do thiếu đầu vào) hoặc nếu nó hợp lệ thì kết quả tập lệnh sẽ luôn giống hệt nhau. Nếu kết quả tập lệnh luôn giống nhau thì không cần lặp lại việc thực thi tập lệnh.

Tất nhiên trực giác là không đủ. Thuộc tính "xác định" đảm bảo tính chính xác của hoạt động tập hợp lại này cần phải được chính thức hóa và sổ cái cần được chứng minh hoặc kiểm tra cẩn thận để đảm bảo rằng nó đáp ứng thuộc tính này.

Phân tích sơ bộ cho thấy rằng tính năng "địa chỉ con trỏ" (Pointer Address) không đáp ứng thuộc tính xác định này và tính năng này sẽ cần phải bị xóa để hỗ trợ Leios. Tính năng địa chỉ con trỏ đã được đưa vào Shelley và nhằm mục đích cho phép các địa chỉ ngắn hơn, nhưng tính năng này tỏ ra khó sử dụng đối với các ví nên rất ít được sử dụng. Do đó, hy vọng việc loại bỏ nó sẽ không gây ra nhiều khó khăn.

4.7 Yêu cầu đối với lớp mạng lưới

Lớp mạng Cardano sẽ cần được mở rộng để hỗ trợ Ouroboros Leios, nhưng kiến trúc cơ bản của lớp mạng sẽ không thay đổi. Bộ "giao thức nhỏ" cụ thể được sử dụng để hỗ trợ thuật toán đồng thuận sẽ cần được mở rộng, nhưng chẳng hạn như

cách thức mà các Node ngang hàng quản lý các kết nối của chúng với nhau trong mạng ngang hàng sẽ vẫn như cũ.

Đặc biệt, trong Ouroboros Leios có một số loại đối tượng mới cần được chuyển tiếp trên mạng: Input Block, Endorsement Block và báo cáo chứng thực. May mắn thay, cách những thứ này cần được truyền đi đều tuân theo một mô hình hiện có: đó là chuyển tiếp các giao dịch, sử dụng bộ đệm trung gian (tức là Mempool). Do đó, thiết kế rõ ràng nhất là có ba giao thức nhỏ mới để chuyển tiếp ba loại đối tượng mới, mỗi loại có bộ đệm tương ứng. Một điểm khác biệt giữa các giao dịch chuyển tiếp, các Block và báo cáo, đó là số lượng các Block và báo cáo hợp lệ bị giới hạn (theo thống kê) bởi lịch trình lãnh đạo dựa trên VRF của chúng, trong khi không có ràng buộc cố hữu nào đối với nhu cầu gửi giao dịch.

Các giao thức đồng bộ hóa chuỗi và tìm nạp Block sẽ vẫn được sử dụng, nhưng chỉ cho các Ranking Block. Tất cả các Block khác theo kiểu chuyển tiếp.

Để bắt kịp trên chuỗi, cũng cần phải có khả năng tìm nạp các loại Block khác: Input Block và Endorsement Block. Giao thức tìm nạp Block sẽ cần được mở rộng một cách phù hợp.

Do thực tế là Leios có thể sử dụng phần lớn hơn nhiều tài nguyên mạng có sẵn, nên có thể cần phải đưa ra khả năng kiểm soát chính xác hơn đối với việc ưu tiên truyền các mạng khác nhau: giữa các loại Block khác nhau và giữa các Node ngang hàng khác nhau.

5 Chiến lược phát triển

Việc triển khai Ouroboros Leios trong Cardano sẽ là một công việc quan trọng và sẽ cần được tiếp cận cẩn thận. Để đáp ứng các mục tiêu về hiệu suất, đồng thời duy trì hoặc cải thiện tính bảo mật và chất lượng chung có thể sẽ đòi hỏi nỗ lực nghiên cứu và phát triển trong nhiều năm.

Nhìn chung, chiến lược phát triển sẽ là chuẩn bị cẩn thận, tạo nguyên mẫu, phân tích và chính thức hóa, trước khi cố gắng tích hợp bất kỳ thứ gì vào cơ sở mã Code hiện có. Chiến lược này sẽ tối đa hóa cơ hội thành công và giảm thiểu khả năng xảy ra sự cố cũng như gây ra sự chậm trễ trong các giai đoạn triển khai và tích hợp sau này.

Dưới đây là một số bước chính liên quan:

- Thiết kế nghiên cứu Ouroboros Leios phải được hoàn thành, bao gồm phân tích bảo mật, được xuất bản và đánh giá ngang hàng.
- Các tham số giao thức khác nhau phải được phân tích dựa trên kết quả bảo mật và được hiệu chỉnh. Điều này có thể liên quan đến công việc thiết kế tiếp theo nếu một số tham số gây khó khăn thực tế.
- Thuật toán Leios cần được mô tả bằng ngôn ngữ khoa học máy tính chính thức phù hợp. Điều này khác với các mô tả được sử dụng trong các tài liệu mật mã học thuật để phân tích bảo mật. Mục đích là để đưa ra một mô tả

chính xác về hành vi của hàm, giúp truyền đạt thiết kế cho nhóm kỹ sư và sau đó là giúp kiểm tra hành vi.

- Hiệu suất là rất quan trọng đối với Ouroboros Leios nên sẽ cần phải thực hiện nhiều phân tích và tạo nguyên mẫu hiệu suất hơn cho Leios so với các thiết kế Ouroboros trước đây. Điều này là để đảm bảo rằng các đặc tính hiệu suất được hiểu đúng và cung cấp một mức độ mô đun hóa trong việc thực hiện thiết kế và triển khai. Mô đun hiệu suất nghĩa là có thể đưa ra các yêu cầu về hiệu suất cho các thành phần của hệ thống mà nếu đáp ứng thì sẽ có nghĩa là toàn bộ hệ thống đáp ứng các yêu cầu về hiệu suất của nó. Điều này rất quan trọng để tìm và khắc phục sớm các vấn đề về hiệu suất, thay vì chỉ phát hiện ra khi điểm chuẩn cấp hệ thống được thực hiện và sau đó gặp khó khăn trong việc xác định thành phần hoặc sự kết hợp của các thành phần là nguyên nhân.
- Việc triển khai nguyên mẫu chi tiết sẽ rất quan trọng để đảm bảo rằng tất cả các chi tiết của thiết kế đều được bao gồm, được hiểu và chúng hoạt động cùng nhau và đạt được các mục tiêu. Nguyên mẫu này sẽ có thể chạy trong mô phỏng và kiểm tra một số hành vi động như sử dụng tài nguyên, tranh chấp tài nguyên và một số yêu cầu về hiệu suất.
- Khi một nguyên mẫu đầy đủ của thiết kế đã được tạo và xác thực, sẽ cần phải lập kế hoạch chi tiết về cách điều chỉnh việc triển khai Cardano hiện có để hỗ trợ Ouroboros Leios, cộng với cách quản lý suôn sẻ một Hardfork từ Ouroboros Praos sang Leios.
- Trong lớp sổ cái, sẽ có rất nhiều công việc chính thức hóa cần thực hiện để đảm bảo rằng sổ cái có thể hỗ trợ chính xác các hoạt động mới.

Tất cả những điều này là cần thiết trước khi công việc triển khai bắt đầu, tất nhiên sẽ mất thời gian. Giai đoạn triển khai cũng là khi các thử nghiệm dựa trên thuộc tính và điểm chuẩn cấp thành phần được phát triển, dựa trên các thuộc tính và yêu cầu về hiệu suất đã xác định trước đó.

Sẽ cần phải có một giai đoạn Testnet mở rộng để cho phép cộng tác với các tác giả của nhiều công cụ và ứng dụng Cardano, để hỗ trợ họ nâng cấp từ Praos lên Leios.

6 Phụ thuộc và mối quan hệ với các tính năng khác

Việc triển khai Ouroboros Leios ở quy mô đầy đủ trong Cardano sẽ phụ thuộc vào một số tính năng khác.

6.1 Lưu trữ trạng thái sổ cái trên ổ đĩa

Có một dự án đang diễn ra để chuyển trạng thái sổ cái của Node Cardano từ hoàn toàn trong bộ nhớ sang được lưu trữ chủ yếu trên ổ đĩa. Điều này là cần thiết để mở rộng sang các trạng thái sổ cái lớn hơn, chẳng hạn như nhiều UTxO hơn, nhiều

người dùng hơn, nhiều tập lệnh hơn được lưu trữ trong UTxO, v.v. Nó cũng sẽ giảm đáng kể yêu cầu RAM cho các Node sau khi được triển khai đầy đủ.

Việc triển khai ban đầu tính năng này sử dụng chương trình phụ trợ trên ổ đĩa có hiệu suất trung bình, phù hợp với mức thông lượng hiện có trên Cardano. Một phụ trợ hiệu suất cao đang được lên kế hoạch. Một biến thể được điều chỉnh phù hợp của điều này sẽ cần thiết cho các yêu cầu thông lượng cao của Ouroboros Leios.

Lưu ý rằng ổ SSD cục bộ sẽ trở thành một phần của yêu cầu tối thiểu đối với Node[12] nhưng yêu cầu về RAM sẽ trở nên tương đối khiêm tốn.

6.2 Ouroboros Genesis

Ouroboros Genesis là một tính năng mô đun cho phép các Node khởi động Blockchain một cách an toàn từ Genesis với các yêu cầu tin cậy tối thiểu. Tính năng này đang được phát triển để cho phép tắt các IOG Relay, cùng với việc triển khai ngang hàng, sẽ hoàn thành việc phi tập trung mạng lưới vật lý.

Cũng giống như Ouroboros Praos+Genesis là một sự kết hợp hợp lệ, Ouroboros Leios+Genesis cũng có ý nghĩa hoàn hảo và mang lại những lợi ích tương tự. Vậy nên theo một nghĩa nào đó, Leios và Genesis là các tính năng độc lập. Trong thực tế, Genesis sẽ được triển khai vào thời điểm Leios khả dụng nên sẽ là một sự thụt lùi nếu không bao gồm tính năng Genesis.

6.3 Chuẩn bị trong sổ cái

Sổ cái Cardano hiện tại phù hợp với các yêu cầu của Ouroboros Leios, nhưng sẽ cần một số công việc chuẩn bị. Ngoài các hoạt động tuần tự hóa mới, các quy tắc sổ cái hiện tại phải đáp ứng thuộc tính xác định mà chúng tôi tin là cần thiết. Tính chất xác định cũng cần phải được chính thức hóa và được sử dụng để chứng minh rằng hoạt động tập hợp lại là chính xác.

Cụ thể, tính năng địa chỉ con trỏ được cho là không thỏa mãn thuộc tính này và tính năng này sẽ cần phải bị xóa. Các quy tắc sổ cái còn lại sẽ cần được kiểm tra để xác minh rằng chúng có đáp ứng thuộc tính xác định hay không.

7 Định hướng tương lai

Thật khó để dự đoán các hướng nghiên cứu và phát triển trong tương lai, nhưng có một số vấn đề đã đủ rõ ràng và đáng được đề cập.

[12]Đây là trường hợp của tất cả các Node nên ví Full-Node cũng vậy.

7.1 Độ trễ giảm

Thiết kế Ouroboros Leios đánh đổi độ trễ tăng lên để tăng thông lượng. Cụ thể, điều này có nghĩa là độ trễ của các giao dịch được đưa vào chuỗi và cả độ trễ giữa các giao dịch (trong các Input Block khác nhau) có thể phụ thuộc lẫn nhau. Một chủ đề nghiên cứu là điều tra các thay đổi đối với thiết kế của Leios nhằm giảm độ trễ tối thiểu giữa các giao dịch phụ thuộc. Cụ thể, một ý tưởng đầy hứa hẹn là thay vì các Input Block tham chiếu Ranking Block trước đó làm ngữ cảnh để xác thực, thay vào đó, họ có thể sử dụng trạng thái sổ cái "quan sát" bằng cách tham chiếu Ranking Block và Endorsement Block. Endorsement Block được tham chiếu phải là Block đã có đủ báo cáo để tạo chứng chỉ xác nhận, điều này đảm bảo rằng nó có sẵn đầy đủ cho tất cả những người tham gia. Ý nghĩa của trạng thái sổ cái là sau đó trạng thái sổ cái của Ranking Block được mở rộng với trạng thái sổ cái được hình thành từ việc tuần tự hóa tất cả các Input Block trong Endorsement Block. Vì các Endorsement Block được tạo thường xuyên hơn so với các Ranking Block nên điều này sẽ làm giảm độ trễ tổng thể giữa các giao dịch phụ thuộc.

7.2 Khả năng mở rộng theo chiều ngang

Thiết kế Ouroboros Leios cung cấp cái được gọi là "khả năng mở rộng theo chiều dọc". Khái niệm về khả năng mở rộng này có nghĩa là khi một người tăng tài nguyên mạng và tính toán của các Node riêng lẻ, thông lượng có thể tăng tương ứng. Ví dụ, theo cách cổ điển, người ta sẽ mô tả các máy chủ cơ sở dữ liệu hoặc máy chủ Web có khả năng mở rộng theo chiều dọc nếu thông lượng của chúng tăng lên khi chạy trên các máy tính "lớn hơn": nhiều Core hơn, Core nhanh hơn và hệ thống I/O nhanh hơn. Trong bối cảnh Cardano, điều này có nghĩa là tăng các yêu cầu hệ thống tối thiểu, ví dụ: có nhiều CPU Core hơn hoặc băng thông mạng cao hơn, sau đó có thể sử dụng các tài nguyên đó để tăng thông lượng, bằng cách tăng tần suất của các Input Block hoặc kích thước của các Input Block.

Tuy nhiên, thiết kế Leios hiện tại không cung cấp "khả năng mở rộng theo chiều ngang". Khái niệm về khả năng mở rộng này có nghĩa là khi một người tăng số lượng máy tính chạy hệ thống, thông lượng có thể tăng lên. Ví dụ: một cụm máy chủ Web hoặc máy chủ cơ sở dữ liệu sẽ được mô tả là có thể mở rộng theo chiều ngang nếu thông lượng của cụm tăng lên khi có nhiều máy tính được thêm vào cụm.

Một hướng nghiên cứu trong tương lai là xem liệu các biến thể hoặc phần mở rộng trên thiết kế của Leios có thể cung cấp một mức độ khả năng mở rộng theo chiều ngang hay không. Có một số dấu hiệu cho thấy điều này có thể xảy ra. Ví dụ: nếu thay vì tất cả các Block Producer xác thực tất cả các Input Block thì họ chỉ xác thực các Input Block mà họ cần báo cáo, và nếu cơ chế báo cáo sẽ phân chia các Input Block xung quanh những người báo cáo khác nhau thì mỗi người báo cáo sẽ chỉ cần nhìn vào một phần nhỏ của các Block. Điều này sẽ cho phép

tăng tổng số Input Block. Kế hoạch này cũng sẽ có lợi ích tích cực là những người nắm giữ cổ phần lớn hơn sẽ được kỳ vọng sẽ xác thực nhiều Block hơn so với những người nắm giữ cổ phần nhỏ. Điều này là công bằng theo nghĩa là những người nắm giữ cổ phần lớn hơn sẽ cần dành nhiều nguồn lực hơn những người nắm giữ cổ phần nhỏ hơn, nhưng tương ứng những người nắm giữ cổ phần lớn hơn có thể mong đợi phần thưởng cao hơn.

Sẽ có những khía cạnh khác của thiết kế cũng cần được sửa đổi để đạt được một thiết kế có thể mở rộng theo chiều ngang. Cụ thể, một số cơ chế sẽ cần được tìm thấy để xử lý các giao dịch ở phía gửi để không phải tất cả các Node đều cần xác thực tất cả các giao dịch.

8 Mainnet Cardano: nhanh như thế nào là quá nhanh?

Nếu Ouroboros Leios có thể được triển khai thành công và nó được triển khai trên mạng chính Cardano thì cộng đồng sẽ phải đối mặt với một câu hỏi mà nó chưa từng phải đối mặt trước đây: làm thế nào để cân bằng sự đánh đổi giữa hiệu suất hệ thống và chi phí tài nguyên khi tham gia. Hiệu suất hệ thống cao hơn cho phép sử dụng hệ thống nhiều hơn, nhưng chi phí tài nguyên cao hơn có thể khiến việc tham gia vượt quá khả năng của nhiều người dùng thông thường. Nói một cách đơn giản: hệ thống có thể quá nhanh (và quá ngốn tài nguyên) khiến hầu hết mọi người không muốn sử dụng nó.

Trong một thế giới hoàn hảo, có thể có thông lượng cao và mức sử dụng tài nguyên thấp cho người tham gia, nhưng trong thế giới thực, có một sự đánh đổi: thông lượng tăng đi kèm với việc sử dụng tài nguyên tăng. Thật vậy, toàn bộ lực đẩy của phương pháp thiết kế Leios là sử dụng đầy đủ hơn các nguồn lực sẵn có như một phương tiện để tăng thông lượng.

Ví dụ: nếu ngân sách thực thi tập lệnh Mainnet được đặt thành một phần đáng kể trong số 4 Core của máy chủ x86 thông thường thì các yêu cầu phần cứng tối thiểu tương ứng cho tất cả các Node xác thực sẽ cần phải là Chip x86 4 Core (hoặc tương đương sức mạnh). Điều này nhất thiết sẽ loại trừ những người đang vận hành Block Producer hoặc Relay trên Raspberry Pis của họ. Quan trọng hơn, nó sẽ làm cho ví Full-Node trở nên không thực tế đối với hầu hết người dùng.

Cộng đồng Cardano không phải là cộng đồng duy nhất đối mặt với câu hỏi này. Chủ đề tương tự cũng xuất hiện ở Bitcoin, mặc dù ở mức hiệu suất thấp hơn. Cộng đồng Bitcoin đã tranh luận về giá trị của các Block lớn và Block nhỏ. Một trong những lập luận ủng hộ các Block nhỏ là "để cung cấp cho người dùng cuối tùy chọn dễ dàng chạy một Node và do đó có một hệ thống phi tập trung hơn"[13].

[13]Trích dẫn Till Musshoff *giải thích về cuộc chiến kích thước khối Bitcoin* https://www.bitrawr.com/bitcoin-block-size-debate-explained

Cần xem xét ngắn gọn việc tăng các yêu cầu về nguồn lực tối thiểu có thể trông như thế nào, hậu quả sẽ ra sao và những biện pháp giảm thiểu nào có thể thực hiện được.

Giả sử để lập luận rằng hệ thống có thể chạy thành công với tốc độ 5 Input Block mỗi giây, với mỗi Input Block là 100kB.

Dung lượng ổ đĩa: Cấu hình ví dụ cho tốc độ dữ liệu là 500kB/s. Với tốc độ này, dung lượng lưu trữ ổ đĩa cho chuỗi sẽ tăng khoảng 41 GB/ngày hoặc 14,7 TB/năm.

Điều này rõ ràng là không thực tế đối với người dùng gia đình để lưu trữ toàn bộ chuỗi. Ổ cứng Multiterabyte không rẻ và ngay cả máy tính để bàn lớn cũng chỉ có thể chứa một vài ổ cứng.

Băng thông mạng: Băng thông mạng thường được đo bằng Megabit trên giây, không phải Byte trên giây. Băng thông cấp ứng dụng 500kB/s có nhiều giao thức khác nhau nên ở mạng cấp thấp sẽ tương ứng với khoảng 5 Mbit/s. Đây sẽ là một phần đáng kể trong hầu hết các kết nối băng thông rộng của người dùng gia đình và đối với nhiều người dùng sẽ vượt quá giới hạn sử dụng.

Bộ nhớ: Với phân tích này, chúng tôi giả định rằng thiết kế hiện tại để lưu trữ trạng thái sổ cái trên ổ đĩa sẽ được hoàn thành. Bộ nhớ chính sẽ được yêu cầu cho các chỉ số và bộ lọc nở hoa (Bloom Filter) cho các bảng lớn trên ổ đĩa, chẳng hạn như UTxO. Các ước tính ban đầu cho thấy rằng một UTxO gồm 100 triệu mục nhập (gần bằng kích thước của Bitcoin) sẽ chỉ cần vài trăm Megabyte bộ nhớ cho các chỉ số và bộ lọc nở hoa. Thiết kế lưu trữ trên ổ đĩa hiện tại liên quan đến việc duy trì sự khác biệt trong các bảng trên ổ đĩa trong bộ nhớ cho K Block cuối cùng, khoảng 12 giờ. Điều này có thể chấp nhận được đối với tốc độ dữ liệu Cardano hiện tại. Tuy nhiên, với tốc độ 500kB/s, nó sẽ cần nhiều bộ nhớ hơn. Một tính toán sơ bộ là 500 kB có thể bao gồm khoảng 1000 giao dịch nhỏ, mỗi giao dịch có 2 UTxO đầu vào và đầu ra, do bộ nhớ đại diện cho các khác biệt có thể cần tới 8 GB cho các chênh lệch có giá trị trong 12 giờ. Điều này gợi ý rằng những thay đổi thiết kế tiếp theo có thể cần thiết để giữ được nhiều trạng thái hơn nữa trên ổ đĩa.

Tài nguyên CPU: Tài nguyên CPU cần thiết sẽ phụ thuộc trực tiếp vào ngân sách được đặt để thực thi tập lệnh. Về nguyên tắc, có thể đẩy mức này đủ cao để có thể bão hòa một số CPU Core. Sau đó, câu hỏi đơn giản là: những yêu cầu hệ thống tối thiểu nào được chấp nhận.

Điều đáng chú ý như đã thảo luận trong Phần 4.4.2 là thiết kế của Leios cho phép các Node không phải là Block Producer đi theo chuỗi mà không cần phải xác thực nội dung của các Input Block. Vì vậy, các yêu cầu tài nguyên CPU cho các Node Relay và người dùng cuối có thể thấp hơn so với các Node tạo Block.

8.1 Giảm nhẹ

Rõ ràng là trong cấu hình thông lượng rất cao, Ouroboros Leios sẽ có các yêu cầu về tài nguyên quá cao đối với hầu hết người dùng cuối, ít nhất là dựa trên kiến trúc hiện tại của ví Full-Node và lưu trữ toàn bộ chuỗi.

Nếu mong muốn cấu hình thông lượng cao như vậy, đại đa số người dùng có thể cần phải chuyển sang ví nhẹ.

Nó cũng gần như chắc chắn là cần thiết đối với hầu hết các Node đầy đủ của người dùng cuối để không lưu trữ toàn bộ lịch sử chuỗi. Bản thân Node không cần toàn bộ lịch sử chuỗi (ngoại trừ hiện tại khi di chuyển định dạng ảnh chụp nhanh trạng thái sổ cái trên ổ đĩa), nhưng một số ứng dụng khách thì cần. Ví dụ: khôi phục ví BIP44 từ cụm từ bảo mật yêu cầu toàn bộ lịch sử chuỗi hoặc chỉ số của tất cả các địa chỉ đã từng được sử dụng trên chuỗi. Ngoài ra, một số sơ đồ sẽ cần được phát triển để hầu hết các Relay cũng không cần lưu trữ toàn bộ chuỗi mà chỉ lưu trữ chuỗi gần đây. Tất nhiên, toàn bộ chuỗi cần được lưu giữ ở đâu đó và sơ đồ như vậy sẽ cần có khả năng hỗ trợ các Node tìm thấy các phần cũ hơn của chuỗi có sẵn ở đâu trong mạng.

Việc đồng bộ hóa chuỗi cũng có thể mất nhiều thời gian. Có thể cần phải xây dựng một tính năng để tạo ảnh chụp nhanh trạng thái chuỗi theo cách mà các Node có thể thiết lập niềm tin vào ảnh chụp nhanh và tiếp tục theo dõi chuỗi từ đó. Một tính năng như vậy hiện đang được tạo mẫu, với tên mã Code là “Mithril”. Đây có thể là một tính năng tích hợp cần thiết để cho phép cấu hình hiệu suất cao.

Mỗi giảm thiểu này là một tính năng không tầm thường theo cách riêng của chúng. Điều này cần được tính đến khi lập kế hoạch tích hợp Ouroboros Leios, nếu muốn có một cấu hình hiệu suất thực sự cao.

Người dịch: Nguyễn Văn Tú
Telegram: https://t.me/Tulibra
Nguồn tài liệu: https://iohk.io/en/research/library/papers/ouroboros-leios-design-goals-and-concepts/