WhitePaper

Giovanni Giorgi
29/10/2017 V.1.1

Abstract

In questo documento viene descritta una nuova tipologia di DLT (distributed ledger technology) che permette una maggiore efficienza di memorizzazione rispetto alla normale implementazione di blockchain, mantenendo le stesse caratteristiche di consenso sulla rete.
Inoltre, grazie all'utilizzo di una struttura ibrida che combina un'implementazione custom della blockchain con un digrafo aciclico connesso, garantisce maggiore scalabilità e prestazioni migliori, senza costi di utilizzo e senza possibilità di accentramento di potere su un sottogruppo di utenti.

Il ledger è pensato oltre che per lo scambio di valuta senza fee, anche per il salvataggio di dati e risorse di varia tipologia, in modo da permettere un buon supporto per l'implementazione di smart contracts e per essere usato come backend per web 3.0

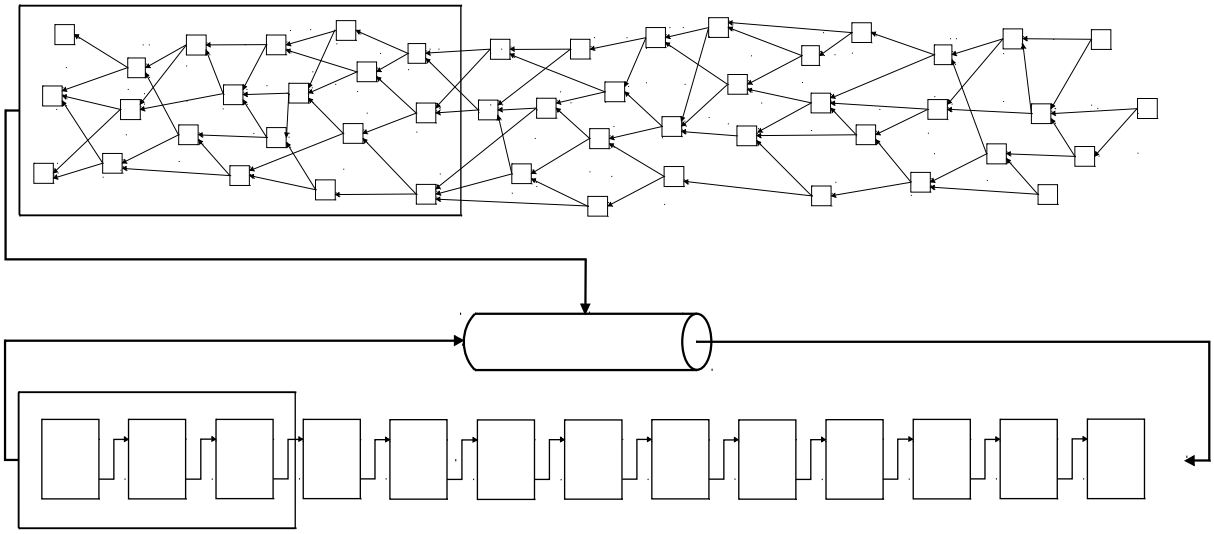
1.1 Introduzione

La tecnologia blockchain è sicuramente una delle più grandi innovazioni degli ultimi anni. La sua introduzione ha permesso di creare un ledger decentralizzato per la gestione trustless di scambio di valuta e dati, garantendo integrità e consistenza su tutti i nodi della rete e rendendo praticamente impossibile la manomissione dei dati inseriti nei blocchi.
Nonostante i notevoli vantaggi, già ampiamente dimostrati dalla diffusione di bitcoin [1] e dalla nascita di numerose alternative basate su tecnologia analoga, la blockchain presenta anche importanti limite dovuti ad una struttura troppo rigida che limita la capacità di gestire un'importante mole di dati e di scalare con l'aumentare dell'utilizzo.

Inoltre il sistema di controllo basato sul lavoro dei miners, oltre ad essere poco efficiente dal punto di vista energetico, permette un accentramento di potere su un ristretto gruppo di utenti, rendendo difficoltoso l'introduzione di nuove funzionalità o l'aggiornamento del sistema, a meno che non venga garantito il guadagno dei miners.
Molti di questi problemi sono risolvibili sostituendo la struttura lineare della blockchain con un DAG (Directed Acyclic Graph) [2]: una struttura più flessibile che permette a chiunque di aggiungere blocchi, senza vincoli di tempistiche prestabilite e senza sottoporre i propri dati al lavoro concorrenziale dei miners e quindi senza necessità di pagare fee.
Questa struttura ha però lo svantaggio di avere un maggiore overhead e di conseguenza una minore efficienza nel mantenimento dei dati sul lungo periodo.

La soluzione è unire le due tecnologie; combinando la scalabilità, assenza di mining e fee tipica del DAG con la sicurezza ed efficienza di memorizzazione della blockchain. Questo tipo di implementazione permette anche di modificare il funzionamento della blockchain, rielaborando il concetto di inalterabilità e rendendo possibile un'ulteriore diminuzione nell'utilizzo di memoria rispetto ai classici ledger usati per criptovalute, pur mantenendo il consenso sulla rete.

1.2 Caratteristiche generali

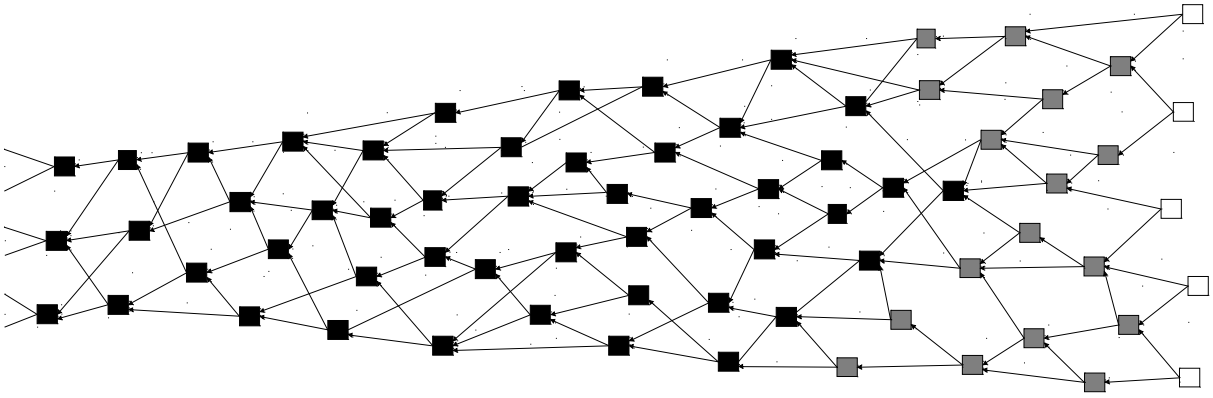


Il ledger prevede l'utilizzo di una struttura ibrida formata da una prima struttura di memorizzazione basata su DAG che permette a chiunque di aggiungere istantaneamente le proprie transizioni, senza necessità di rivolgersi ad un miner e quindi senza necessità di pagare fee.
Questo componente gestisce anche l'approvazione delle transizioni e la sicurezza, con un sistema tale per cui ogni nuova transizione approva un sottogruppo di transizioni precedenti, in modo che le prestazioni del sistema aumentino con l'aumento dell'utilizzo (al contrario della blockchain, dove le prestazioni diminuiscono).
Con l'aumento dell'utilizzo aumenterà anche la sicurezza perché il peso della transizione (concetto spiegato di seguito) aumenterà più velocemente, rendendo più efficiente la gestione di double spending e rendendo più difficili da manipolare i dati aggiunti.

Per garantire un salvataggio in memoria efficiente e sicuro sul lungo periodo, a intervalli regolari, le transizioni più vecchie del DAG, verranno prelevate e salvate sulla seconda struttura di memorizzazione: la Shifting BlockChain.
questa blockchain, invece che memorizzare direttamente le transizioni, contiene un database formato da coppie chiave-valore; in modo da permettere il salvataggio dei balance relativi ad ogni account ed eventualmente altre tipologie di dato e asset.
A differenza delle normali blockchain, i dati non sono completamente inalterabili, ma è prevista una fase di rielaborazione che permette di eliminare i blocchi più vecchi, aggiornarli e aggiungerli nuovamente in testa.
L'aggiornamento segue regole prestabilite ed è sincrono su tutti i nodi della rete, in modo da mantenere il consenso sui dati presenti sul database.

2.1 DAG (Directed Acyclic Graph)

Il DAG è un grafo orientato formato da blocchi, che a differenza della blockchain non seguono un ordine sequenziale, in modo da poter aggiungere blocchi in parallelo, senza conoscere l'intero grafo.
I blocchi contengono esclusivamente le transizioni eseguite da un singolo individui e possono essere aggiunte direttamente dall'utente stesso, dopo aver eseguito il PoW (per proteggere la rete dallo spam e aumentare la sicurezza)e vengono collegati a 2 blocchi precedenti, in modo da convalidarli e renderne più difficile la manomissione.
Per ogni blocco è possibile individuare una profondità (depth) che rappresenta il numero di blocchi presenti nel percorso più lungo per raggiungere un blocco senza conferme (tip); un peso(weight) che ha un certo valore iniziale basato sul PoW eseguito su quel blocco, a cui vengono sommati i pesi dei blocchi collegati di seguito e un punteggio(score) rappresentato dal peso iniziale del blocco sommato agli score dei blocchi che conferma.



2.2 Aggiunta di nuovi blocchi e convalida

Per aggiungere un nuovo blocco, l'utente deve selezionare due blocchi precedenti (ancora non confermati) per convalidarli.
il processo di convalida consiste nel certificare la validità della firma, che il contenuto non sia in conflitto con blocchi precedenti e che sia stato eseguito il pow.
se i blocchi selezionati sono legittimi possono essere usati per agganciare il nuovo blocco.

Con l'aggiunta di un blocco, si conferma solo la sezione di grafo sequenzialmente precedente ai blocchi direttamente selezionati, questo comporta un aumento del peso dei blocchi in quella sezione.
Un blocco viene considerato completamente confermato quando tutte le sezioni di grafo seguenti lo confermano, ovvero tutti i nuovi blocchi (tip) ne aumentano il peso.
Tutti i blocchi che hanno una adeguata profondità sono completamente confermati.

per selezionare i blocchi da approvare viene utilizzato un algoritmo randomico che da maggiore probabilità ai blocchi con uno score più elevato, in modo da incentivare gli utenti ad eseguire un pow con difficoltà maggiore e rendendo più sicura la rete e limitando la divergenza del grafo, che provocherebbe una diminuzione delle prestazioni.

2.3 Gestione conflitti

In caso di un conflitto può capitare che entrambi i blocchi siano in un primo momento approvati in sezioni differenti del DAG ma è impossibile che un singolo blocco approvi una sezione di grafo che le confermi entrambe contemporaneamente.
anche se il conflitto non viene individuato immediatamente, con il passare del tempo la sezione di grafo con il blocco maligno con il punteggio minore verrà abbandonato e verrà portata alla convalida completa solo uno dei due blocchi; come effetto collaterale tutti i blocchi che si sono agganciati di seguito non verranno mai confermati.
per gli utenti è possibile agganciarli nuovamente in una posizione differente senza modifiche, a parte eseguire nuovamente il pow.

È possibile continuare ad aggiungere blocchi anche se si perde la connessione degli altri nodi ma con il rischio che il nuovo tratto non sia conforme al resto della rete.
Nel caso ci siano problemi di approvazione, è necessario collegare i blocchi nuovamente in una differente posizione.

3.1 Shifting BlockChain:

Per il salvataggio nel lungo periodo viene utilizzata un'implementazione custom della blockchain.

La struttura del blocco è quella classica; contenente oltre alla sezione dedicata ai dati, un numero identificativo sequenziale e l'hash del blocco precedente.
Manca la sezione dedicata al PoW perché la sicurezza viene gestita separatamente, in modo che l'aggiunta di nuovi blocchi non sia vincolata da tempistiche prestabilite.
Il blocco non ha limiti pratici di dimensione, questo è possibile perché in fase di aggiunta, tutti i dati da inserire sono già presenti nel nodo e non bisogna tener conto dei problemi di propagazione del contenuto tra i nodi.

A differenza delle classiche blockchain viene introdotto il concetto di aggiornamento dello stato dei dati.
A intervalli regolari (fase di Add & Update) vengono prelevati i blocchi in coda e eliminati dalla blockchain; i dati prelevati vengono riorganizzati, integrati con i nuovi dati da inserire e vengono eliminati i record ridondanti.
In seguito il tutto viene aggiunto in testa.

La sicurezza viene garantita salvando l'hash dell'ultimo blocco nel DAG per garantire l'inalterabilità e per propagare l'hash tra i nodi, in modo da garantire la consistenza dei dati sulla rete.
Questa operazione viene eseguita solo da una certa percentuale di nodi; la percentuale viene decisa in base al numero di agganci eseguiti in precedenza e calcolata in modo da garantire una buona sicurezza, senza eccedere con lo spam sulla rete.
Per garantire che la percentuale venga rispettata ogni nodo esegue autonomamente un algoritmo basato sul random per decidere se eseguire questo compito o meno.

Nel caso in cui un nodo si rende conto di avere uno stato differente dagli altri, inizia un colloquio con i nodi vicini per recuperare il giusto stato.
Durante questa fase vengono scambiati gli hash dei vari blocchi per ricostruire la giusta sequenza dei dati e nel caso sia necessario vengono recuperati i blocchi mancanti, in modo da ottenere uno stato conforme con quello del resto della rete.

La perdita di dati dalla coda della blockchain viene impedita dall'impossibilità di ricostruire i nuovi blocchi, senza conoscere il contenuto degli ultimi.

3.2 Add & Update:

Ogni nodo in autonomia, a intervalli regolari, vincola la sezione più vecchia del DAG.
la giusta sezione viene individuata ogni volta che un blocco supera un certo valore di depth e viene selezionata tutta la sezione di blocchi il cui depth rientra in un certo range di valori (D > DM - L)
Il valore di depth che innesca la fase di add and update varia dinamicamente in modo che l'operazione venga eseguita con la giusta frequenza e il range varia in modo che il DAG non assumi dimensioni troppo grandi (che ne rendano difficile la gestione) o troppo piccole (che ne limiti sicurezza e affidabilità).

Eventuali problemi nella propagazione dei blocchi non influenzano l'individuazione della sezione, perchè nel caso venga aggiunto un nuovo blocco, la profondità di tutti i blocchi confermati aumenta dello stesso valore, pertanto la differenza con il blocco con profondità più alta rimane costante.
l'unica cosa che varia da un nodo all'altro è l'istante in cui si inizia il processo.

sui blocchi di questa sezione vengono estratti i record e ordinati in base all'hash
Dalla coda della blockchain viene prelevato un blocco, vengono eliminati i dati presenti in forma aggiornata in una sezione differente di blockchain, vengono aggiornati i dati presenti ed integrati con i dati provenienti dal DAG, fino al raggiungimento di una dimensione prestabilita del blocco.

Il blocco così ottenuto viene spostato in testa alla blockchain,
Se sono presenti altri dati provenienti dal DAG, non aggiunti alla blockchain, viene estratto un nuovo blocco dal fondo e viene ripetuta l'operazione N volte fino a quando non vengono inseriti tutti i dati provenienti dal dag o fino a quando N supera un certo valore NM, in quel caso i dati rimanenti vengono inseriti in un nuovo blocco.

Al termine dell'operazione l'hash dell'ultimo blocco viene salvato nel DAG seguendo la procedura precedentemente descritta.

4.1 Struttura dei dati

Sia nella blockchain che nel dag i blocchi sono formati da record che contengono coppie chiave-valore che seguono la struttura (Chiave, Valore, Address, firma).
Ogni chiave viene associata definitivamente all'address che per primo gli assegna un valore, di default ogni chiave non ammette duplicati e il valore può essere alterato solo firmando un nuovo record per la stessa chiave usando lo stesso address.

come valore di una chiave, aggiungendo il carattere speciale underscore ("-") all'inizio del nome, può essere associato un testo particolare che seguendo uno specifico linguaggio DDL (Data Definition Language) definisce il comportamento di quella chiave e ne definisce le caratteristiche:

1. estendere della chiave a più address, in modo che ogni address possa definire una propria istanza.
2. vincolare la tipologia di dato.
3. imporre l'inalterabilità
4. permettere la struttura ad array, permettendo duplicati della stessa chiave, per lo stesso address, mettendo come suffisso l'indice tra parentesi quadre ("[", "]")

mediante l'utilizzo del DDL è possibile anche definire nuovi token, in quel caso la tipologia di dato sarà UINT e deve essere definito il massimo supply, che verrà inizialmente attribuito all'address che ha creato il token, nel caso dei token il balance può essere aggiornato anche firmando con un address diverso, ma solo se viene aggiunta una quantità positiva, in questo caso nella blockchain la seziona della firma rimane null.

nei blocchi del dag è possibile aggiungere n record (n prestabilito), a patto che non si superi una certa dimensione prestabilita o un unico record se la dimensione viene superata.
nel blocco può essere inserito un record con una chiave riservata che permette di aggiungere un messaggio al blocco, questo record verrà comunque firmato ma non verrà salvato nella blockchain, quindi può essere conservato a descrizione dell'utente

4.2 Transizioni di Token

nei blocchi del dag è possibile aggiornare i balance dei vari address, a patto che nel singolo blocco la somma della quantità tolta da un address sia pari alla quantità aggiunta ad altri e che si aggiorni di una quantità negativa solo l'address con cui si firma.
la quantità tolta va utilizzata per aggiornare uno o più address differenti e tutti i record vanno firmati con la chiave dell'address a cui sono stati tolti i token.

se un utente vuole firmare una transizione che combini i balance di due address verso un terzo in modo che risulti un'unica transizione, può firmare una transizione da un conto di sua proprietà verso l'altro e successivamente creare un nuovo blocco agganciato al primo (che lo conferma) che invii il la somma desiderata al terzo address.
in questo caso, anche si il primo blocco non è stato confermato durante la firma del secondo, il secondo può comunque inviare l'intera somma perché se il secondo blocco viene approvato, allora viene approvato anche il primo, visto che sono collegati

References:

1. S. Nakamoto (2008) “Bitcoin: A Peer-to-Peer Electronic Cash System”. https://bitcoin.org/bitcoin.pdf
2. Sergio Demian Lerner (2015) DagCoin: a cryptocurrency without blocks. https://bitslog.wordpress.com/2015/09/11/dagcoin/
3. Serguei Popov (2016) “The Tangle”. https://iota.org/IOTA_Whitepaper.pdf
4. Iota: a cryptocurrency for Internet-of-Things. http://www.iota.org/, and https://bitcointalk.org/index.php?topic=1216479.0