

Technology Know-How: Hypercore Protocol

J. Corina

Corso di laurea magistrale in Ingegneria e Scienze Informatiche
Università di Bologna



- 1 Hypercore Protocol
- 2 Hypercore
- 3 Hyperswarm
- 4 Corestore
- 5 Hyperbee
- 6 Hyperdrive

1 Hypercore Protocol

2 Hypercore

3 Hyperswarm

4 Corestore

5 Hyperbee

6 Hyperdrive

Hypercore Protocol (ora Holepunch)

- Libreria framework che fornisce un insieme di moduli volti a creare reti peer-to-peer per lo scambio di dati
- Molteplici moduli e tool ausiliari associati ad essi
- Disponibile in linguaggio Javascript, necessita dell'ambiente di esecuzione Node.js

1 Hypercore Protocol

2 **Hypercore**

3 Hyperswarm

4 Corestore

5 Hyperbee

6 Hyperdrive

- Implementazione di un log binario **append-only locale**
 - Array di byte, detti **blocchi**, concatenati tra loro
 - Immutabilità
- Struttura dati Merkle tree
 - Verifica dell'integrità con algoritmo Blake2B
 - Versionamento
 - Cifratura con chiave asimmetrica
- Replica attraverso uso di socket o **Hyperswarm**
 - Creazione *discovery key*
- Gestione dei metadati, richiesta del contenuto *on-demand* e *sparse downloading*

1 Hypercore Protocol

2 Hypercore

3 **Hyperswarm**

4 Corestore

5 Hyperbee

6 Hyperdrive

- Modulo che permette ai peer di mettersi in contatto, conoscendo solo la *chiave pubblica* (*Noise public key*) associata ad essi
- Basata sul sotto-componente HyperDHT, sono presenti nodi iniziali di bootstrap
- Condivisione di core (e suoi derivati) attraverso l'uso di
 - *topic*: identificativo di 32 byte
 - associato a *discovery key*
- Gerarchia elastica all'interno dello swarm

1 Hypercore Protocol

2 Hypercore

3 Hyperswarm

4 Corestore

5 Hyperbee

6 Hyperdrive

- Utilizzato come repository e manager di molteplici core presenti sul dispositivo, gestisce logiche di storage e replicazione
- Interfaccia per moduli che lo usano in composizione, come Hyperdrive
- nomi **locali** utilizzati per generare le chiavi di un core, possibile utilizzo di namespaces
- Come identificativo uso la *discovery key* di un core d'appoggio che contiene le chiavi pubbliche dei restanti

1 Hypercore Protocol

2 Hypercore

3 Hyperswarm

4 Corestore

5 Hyperbee

6 Hyperdrive

- Utilizzo della struttura dati **B-Tree** e Hypercore
- Implementazione di un database chiave-valore
 - primitive put,get,del
- Disponibilità di un ottimizzatore **batch** per effettuare operazioni massive
 - *lock*
- Funzioni per gestire la validità di inserimenti/eliminazioni
 - **Compare and swap (CAS)**
- Confronto tra versioni e disponibilità dello storico dei cambiamenti
- Possibilità di creare sub-database (namespaces)
 - *sub-prefix*

- 1 Hypercore Protocol
- 2 Hypercore
- 3 Hyperswarm
- 4 Corestore
- 5 Hyperbee
- 6 Hyperdrive**

Hyperdrive

- Implementazione di un file system distribuito, **non-standalone**
- Utilizzo di Hyperbee per i metadati e Hypercore per il contenuto
- Utilizzo tramite API
 - Interazione con il file system locale tramite API file system di Javascript oppure uso del sotto-componente Localdrive
 - Mirrordrive: sincronizzazione tra file presenti in Localdrive e Hyperdrive
- Versionamento dei cambiamenti e confronto tra versioni
- Accesso tramite chiave-valore
 - chiave: path del file
 - valore: tupla di 4 indici (struttura dati Hyperblob) utili ad accedere al core
 - byteOffset : offset da utilizzare nel blocco iniziale;
 - blockOffset : blocco dal quale iniziare la lettura;
 - blockLength : numero di blocchi da leggere;
 - byteLength : numero totale di byte da leggere.



Grazie per l'attenzione