Law, Science and Technology
MSCA ITN EJD n. 814177

RIDE
Rights of Internet of Everything

Mirko Zichichi[1,2], Luca Serena[2], Stefano Ferretti[3], Gabriele D'Angelo[2]

[1]Universidad Politécnica de Madrid
[2]University of Bologna
[3]University of Urbino "Carlo Bo"

Governing Decentralized Complex Queries Through a DAO

Overview

1. Introduction
2. Hypercube DHT
3. DAO Framework
4. Conclusion

Introduction

General Problem → Possible Solution

General Problem → *data management and delivery*

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control
- affects privacy and access to essential information

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control
- affects privacy and access to essential information
- e.g. Turkey denied the access to the Turkish Wikipedia in 2017

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control
- affects privacy and access to essential information
- e.g. Turkey denied the access to the Turkish Wikipedia in 2017

Possible Solution → *Distributed Ledger Technologies and Decentralized File Storages*

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control
- affects privacy and access to essential information
- e.g. Turkey denied the access to the Turkish Wikipedia in 2017

Possible Solution → *Distributed Ledger Technologies and Decentralized File Storages*

- increasingly used to create **common, decentralized** and **trustless** infrastructures

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control
- affects privacy and access to essential information
- e.g. Turkey denied the access to the Turkish Wikipedia in 2017

Possible Solution → *Distributed Ledger Technologies and Decentralized File Storages*

- increasingly used to create **common, decentralized** and **trustless** infrastructures
- high data **availability**, but also **integrity**, **auditability**, **confidentiality**

General Problem → Possible Solution

General Problem → *data management and delivery*

- single point of failure and arbitrary control
- affects privacy and access to essential information
- e.g. Turkey denied the access to the Turkish Wikipedia in 2017

Possible Solution → *Distributed Ledger Technologies and Decentralized File Storages*

- increasingly used to create **common, decentralized** and **trustless** infrastructures
- high data **availability**, but also **integrity**, **auditability**, **confidentiality**
- ability to **automate and enforce** processes (through smart contracts)

Specific Problem

1) data stored in DLTs and DFS are usually **unstructured** and need to be **filtered and indexed** before any **complex query**

Specific Problem

1) data stored in DLTs and DFS are usually **unstructured** and need to be **filtered and indexed** before any **complex query**

2) there are **no diffused efficient mechanisms to query** a certain type of data, that do not involve **centralization** (e.g. index data in a central database)

Our work

- **Distributed Hash Table (DHT)** $\rightarrow$ distributed data structure that maps "**keys**" into "**values**".

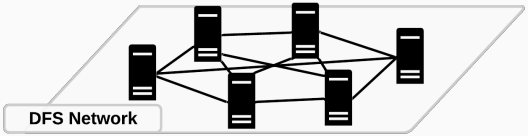
Our work

- **Distributed Hash Table (DHT)** → distributed data structure that maps "**keys**" into "**values**".
- A **Decentralized Autonomous Organization (DAO)** → **smart contracts** to manage rewards and organizational decisions.

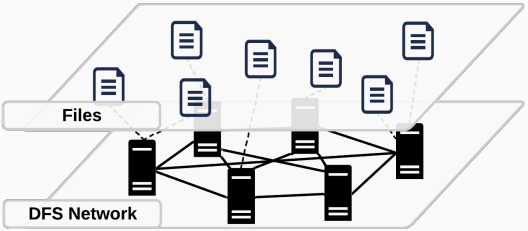
Hypercube DHT

Multiple Keyword Search

DFS **P2P** network → IPFS is using **Content Based Addressing**, i.e. items are directly queried through the network rather than establishing a connection with a server

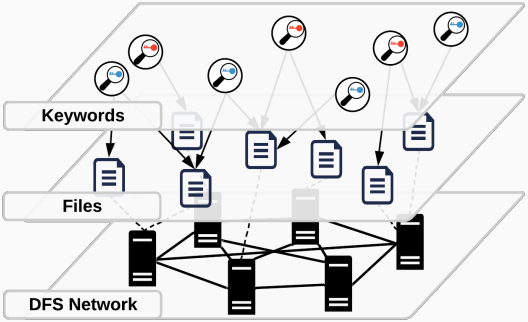


Multiple Keyword Search



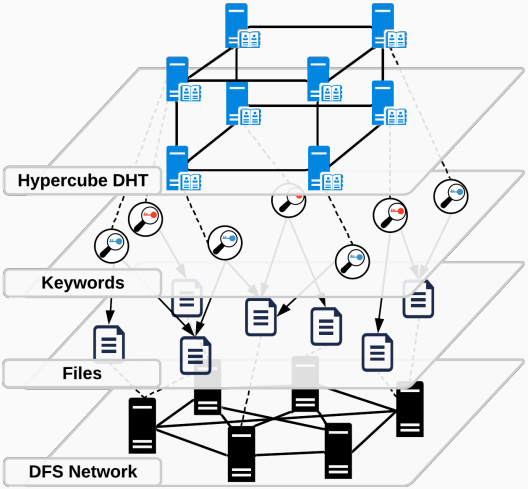
The P2P network that runs the IPFS protocol, stores and shares files in the form of **IPFS objects** that are identified by a **Content IDentifier (CID)**, obtained through an **hash** function.

Multiple Keyword Search



We can **map keywords** to any **IPFS Object** $o \in O$ using a keywords set $K_o \subseteq W$ (keyword space W).

Multiple Keyword Search



A DHT can be exploited to perform **multiple keyword based queries**. In particular one that takes the form of a r-dimensional **hypercube** $H_r(V, E)$.

Keywords Sets

- $O \leftarrow$ set of all the CIDs in IPFS

Keywords Sets

- $O \leftarrow$ set of all the CIDs in IPFS
- $o \in O$ is mapped to a **keyword set** $K_o \subseteq W$

Keywords Sets

- **O** $\leftarrow$ set of all the CIDs in IPFS
- **o** $\in O$ is mapped to a **keyword set** $\mathbf{K}_o \subseteq W$
- By using a **uniform hash function**
  $h : W \rightarrow \{0, 1, \ldots, r-1\}$
  $\mathbf{K}_o$ can be represented by a string of bits **u** $\rightarrow$ 101001

Keywords Sets

- **O** $\leftarrow$ set of all the CIDs in IPFS
- **o** $\in O$ is mapped to a **keyword set** $\mathbf{K}_o \subseteq W$
- By using a **uniform hash function**
  $h : W \rightarrow \{0, 1, \ldots, r-1\}$
  $\mathbf{K}_o$ can be represented by a string of bits **u** $\rightarrow$ 101001
- in **u the 1s are set in the positions** given by
  $one(u) = \{h(k) \mid k \in K\}$

Keywords Sets

- **O** $\leftarrow$ set of all the CIDs in IPFS
- **o** $\in O$ is mapped to a **keyword set** $\mathbf{K}_o \subseteq W$
- By using a **uniform hash function**
  $h : W \rightarrow \{0, 1, \ldots, r-1\}$
  $\mathbf{K}_o$ can be represented by a string of bits **u** $\rightarrow$ 101001
- in **u the 1s are set in the positions** given by
  $one(u) = \{h(k) \mid k \in K\}$
- E.g.: $o = QmbW...MnR$, $K = \{\text{“Wikipedia, Rome”}\}$
  $h(Wikipedia) = 3, h(Rome) = 5$
  K is represented by $u = 000101 \Rightarrow$ **DHT stores (000101,*QmbW...MnR*)**

Hypercube based DHT

- We use these *r*-bit strings to identify logical nodes in a *r*-dimensional **hypercube based DHT**

Hypercube based DHT

- We use these r-bit strings to identify logical nodes in a r-dimensional **hypercube based DHT**
- network topology $\rightarrow$ $H_r(V, E)$ **hypercube**

Hypercube based DHT

- We use these r-bit strings to identify logical nodes in a r-dimensional **hypercube based DHT**
- network topology $\rightarrow$ $H_r(V, E)$ **hypercube**
- **V** set of vertices that represent **logical nodes**

Hypercube based DHT

- We use these r-bit strings to identify logical nodes in a r-dimensional **hypercube based DHT**
- network topology $\rightarrow$ $H_r(V, E)$ **hypercube**
- **V** set of vertices that represent **logical nodes**
- **E** set of edges formed when two vertices differ of only one bit (they are also network **neighbors**), e.g. 1011 and 1010.

Keywords Queries

- **Pin Search** - $\{o \in O \mid K_o = K\}$
  gets all and only the objects associated with a keyword set K
  e.g. *pinSearch({Wikipedia, Rome})* = (*000**1**01*,*QmbW...MnR*), (*000**1**01*,*QmbP...3Lx*), ...

Keywords Queries

- **Pin Search** - $\{o \in O \mid K_o = K\}$
  gets all and only the objects associated with a keyword set K
  e.g. *pinSearch({Wikipedia, Rome})* = (*000**101**,QmbW...MnR*), (*000**101**,QmbP...3Lx*), ...
- **Superset Search** - $\{o \in O \mid K_o \supseteq K\}$
  also gets objects that can be described by keywords sets that include K
  e.g. *superSetSearch({Wikipedia, Rome})* = (*000**101**,QmbW...MnR*),
  (*000**111**,QmbZ...aaD*), ...
  with a limit l of objects.

Setup

- **Hypercube DHT software implemented in Python** → Insert object, Remove object, Pin search, Superset search.

Setup

- **Hypercube DHT software implemented in Python** → Insert object, Remove object, Pin search, Superset search.
- We tested our implementation running the sw on a dedicated host (i.e. a quad core CPU, 16GB RAM), by associating several logical nodes to different OS ports.

Setup

- **Hypercube DHT software implemented in Python** $\rightarrow$ Insert object, Remove object, Pin search, Superset search.
- We tested our implementation running the sw on a dedicated host (i.e. a quad core CPU, 16GB RAM), by associating several logical nodes to different OS ports.
- **Nodes number** $\rightarrow$ from 8 ($r = 3$) up to 128 ($r = 7$)

Setup

- **Hypercube DHT software implemented in Python** → Insert object, Remove object, Pin search, Superset search.
- We tested our implementation running the sw on a dedicated host (i.e. a quad core CPU, 16GB RAM), by associating several logical nodes to different OS ports.
- **Nodes number** → from 8 ($r = 3$) up to 128 ($r = 7$)
- Randomly created keywords-objects → **objects number 10, 100 and 1000**

Setup

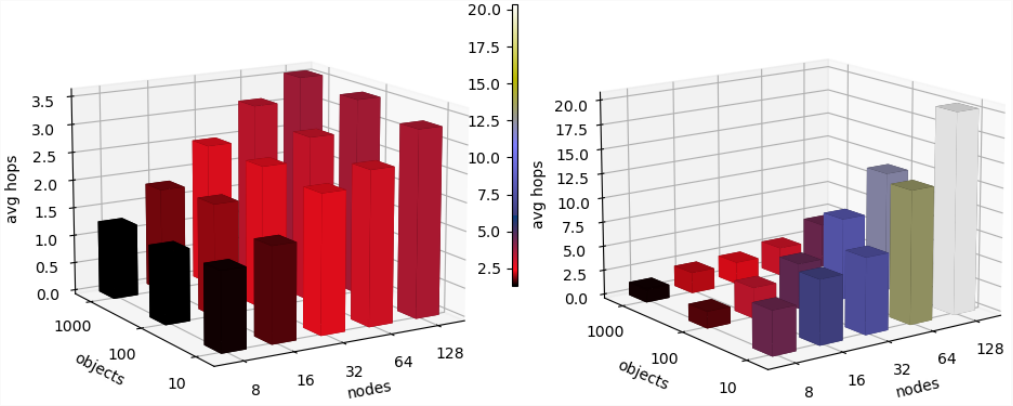
- **Hypercube DHT software implemented in Python** → Insert object, Remove object, Pin search, Superset search.
- We tested our implementation running the sw on a dedicated host (i.e. a quad core CPU, 16GB RAM), by associating several logical nodes to different OS ports.
- **Nodes number** → from 8 ($r = 3$) up to 128 ($r = 7$)
- Randomly created keywords-objects → **objects number 10, 100 and 1000**
- We evaluated the **number of hops** required for each new query

Setup

- **Hypercube DHT software implemented in Python** → Insert object, Remove object, Pin search, Superset search.
- We tested our implementation running the sw on a dedicated host (i.e. a quad core CPU, 16GB RAM), by associating several logical nodes to different OS ports.
- **Nodes number** → from 8 ($r = 3$) up to 128 ($r = 7$)
- Randomly created keywords-objects → **objects number 10, 100 and 1000**
- We evaluated the **number of hops** required for each new query
- For each type of test → 50 repetitions

Pin Search

Superset Search



- order of the **logarithm of the hypercube logical nodes number** $\rightarrow \frac{\log(n)}{2} = \frac{r}{2}$

- plus the average hops to get from the **responsible node** to all the nodes that that include *K*, until the **limit** *l* is reached

DAO Framework

Smart Contracts and Decentralized Autonomous Organizations

- **Smart contracts** → programs whose execution is performed in a distributed way.

Smart Contracts and Decentralized Autonomous Organizations

- **Smart contracts** $\rightarrow$ programs whose execution is performed in a distributed way.
- **Ethereum** $\rightarrow$ nodes receive **same inputs** and perform a computation on the basis of a **contract code** that leads to the **same outputs**.

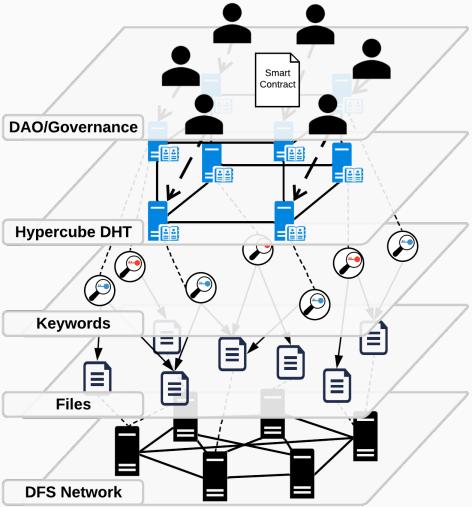
Smart Contracts and Decentralized Autonomous Organizations

- **Smart contracts** → programs whose execution is performed in a distributed way.
- **Ethereum** → nodes receive **same inputs** and perform a computation on the basis of a **contract code** that leads to the **same outputs**.
- Smart contracts can be used to **automatize** and **supervise** the exchange of digital or physical assets, e.g. **tokens**, and to allow the management of a DAO.

Smart Contracts and Decentralized Autonomous Organizations

- **Smart contracts** → programs whose execution is performed in a distributed way.
- **Ethereum** → nodes receive **same inputs** and perform a computation on the basis of a **contract code** that leads to the **same outputs**.
- Smart contracts can be used to **automatize** and **supervise** the exchange of digital or physical assets, e.g. **tokens**, and to allow the management of a DAO.
- **Decentralized Autonomous Organizations** (DAO) → members can make proposals and also vote those through transparent mechanisms.

DAO Framework → Fifth Layer



DAO Framework

- **Token economy** - a unique **ERC20 token** used for transferring **value** within the DAO (e.g. users that pay node operators), or for **staking** purposes.

DAO Framework

- **Token economy** - a unique **ERC20 token** used for transferring **value** within the DAO (e.g. users that pay node operators), or for **staking** purposes.
- **Members Registry** - Any account holding any amount of token can lock some of (or all of) these tokens for a desired amount of time through a specific time-lock contract.

DAO Framework

- **Token economy** - a unique **ERC20 token** used for transferring **value** within the DAO (e.g. users that pay node operators), or for **staking** purposes.
- **Members Registry** - Any account holding any amount of token can lock some of (or all of) these tokens for a desired amount of time through a specific time-lock contract.
- **General Voting** - This contract allows any member to make a **proposal** and gives everyone the opportunity to submit a suggestion to **vote** regarding that proposal. A member **vote weight** is proportional to the amount of tokens locked.
  For instance, DAO members can vote to transfer some staked tokens to a specific account in the case of issuing a bounty.

Use Cases

- **DeFi-based rewarding** - Using a DeFi protocol such as Uniswap, the DAO's token can be automatically exchanged and new Liquidity Pool (LP) tokens will be minted. By locking these LP tokens, the DAO members enable the growth of the token in value and credibility (as a form of auto-financing).

Use Cases

- **DeFi-based rewarding** - Using a DeFi protocol such as Uniswap, the DAO's token can be automatically exchanged and new Liquidity Pool (LP) tokens will be minted. By locking these LP tokens, the DAO members enable the growth of the token in value and credibility (as a form of auto-financing).
- **Unique DAO vs. DAO islands**
  - unique DHT network is governed by a single DAO, with the purpose of assisting different DFS, DLTs and other storages
  - a multitude of DAO "islands" are created for keywords-based queries for specific topics or platforms.

Use Cases

- **DeFi-based rewarding** - Using a DeFi protocol such as Uniswap, the DAO's token can be automatically exchanged and new Liquidity Pool (LP) tokens will be minted. By locking these LP tokens, the DAO members enable the growth of the token in value and credibility (as a form of auto-financing).
- **Unique DAO vs. DAO islands**
  - unique DHT network is governed by a single DAO, with the purpose of assisting different DFS, DLTs and other storages
  - a multitude of DAO "islands" are created for keywords-based queries for specific topics or platforms.
- **Decentralized IPFS-search** - By monitoring the file addition logs, these nodes can download these newly added files and extract the metadata in order to obtain keywords that are then stored in the hypercube DHT with the associated CID.

Gas cost

Smart Contract	Operation	Cost (gas)
ERC20	transfer()	51167
TokenTimelockProxy	lockTokens()	232024
TokenTimelock	release()	25626
Voting	submitProposal()	133501
Voting	submitSuggestion()	114523
Voting	vote()	142848
Voting	executeProposal()	56991

Table 1: DAO smart contracts operations cost in terms of gas.

Conclusion

Conclusion

- **Hypercube DHT** $\rightarrow$ decentralized system that manages **keyword-based queries** for contents stored in IPFS (and not only).

Conclusion

- **Hypercube DHT** $\rightarrow$ decentralized system that manages **keyword-based queries** for contents stored in IPFS (and not only).
- Efficient **trade-off between memory space and response time** $\rightarrow$ maximum number of hops of $\log(\textit{number of nodes}) = r$, i.e. the hypercube dimension.

Conclusion

- **Hypercube DHT** $\rightarrow$ decentralized system that manages **keyword-based queries** for contents stored in IPFS (and not only).
- Efficient **trade-off between memory space and response time** $\rightarrow$ maximum number of hops of $\log(\textit{number of nodes}) = r$, i.e. the hypercube dimension.
- **DAO** $\rightarrow$ related to the economic sustainability and development of the above system.

Conclusion

- **Hypercube DHT** $\rightarrow$ decentralized system that manages **keyword-based queries** for contents stored in IPFS (and not only).
- Efficient **trade-off between memory space and response time** $\rightarrow$ maximum number of hops of $\log(\textit{number of nodes}) = r$, i.e. the hypercube dimension.
- **DAO** $\rightarrow$ related to the economic sustainability and development of the above system.
- DAO ERC20 tokens allows to reward nodes that have actively contributed.

Conclusion

- **Hypercube DHT** → decentralized system that manages **keyword-based queries** for contents stored in IPFS (and not only).
- Efficient **trade-off between memory space and response time** → maximum number of hops of $\log(\textit{number of nodes}) = r$, i.e. the hypercube dimension.
- **DAO** → related to the economic sustainability and development of the above system.
- DAO ERC20 tokens allows to reward nodes that have actively contributed.
- Future works → "pay-per-query" model and load balancing with a more realistic content distribution.