Manenti Edoardo

Simone Nicosanti

Andrea De Filippis

SDCC PROJECT 2K23

A.A. 2022/2023

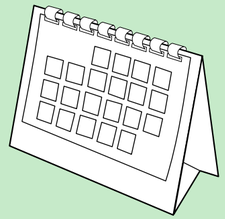
STORAGE NEL CLOUD CONTINUUM

Presentazione del progetto

“SAE storage service”

Let's get started

AGENDA



Introduzione

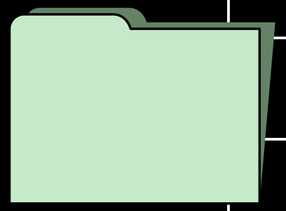
Architettura

Scelte implementative

Risultati empirici

Sviluppi futuri

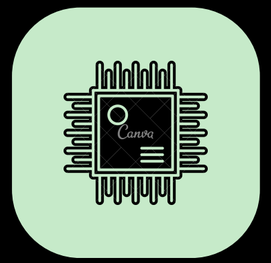
INTRODUZIONE



In questa presentazione entreremo nei dettagli di *SAE Storage System* sviluppato come progetto per l’esame di Sistemi Distribuiti.

Per prima cosa vediamo i building blocks e l’architettura di base...

BUILDING BLOCKS



EDGE

Nodo con risorse limitate e capacità di caching che comunica con il cloud ed altri dispositivi a formare una rete non strutturata che costituisce un layer intermedio tra i client e il cloud.

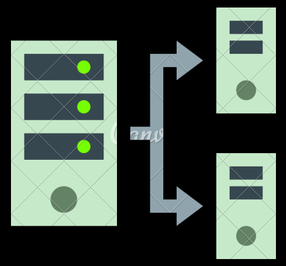
CLIENT

Client che, una volta autenticato, può richiedere di inserire o eliminare file nella rete.

S3 CLOUD

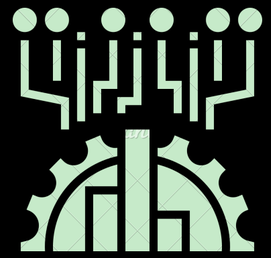
Bucket S3 su amazon web services su cui sono salvati i file dell’intero sistema.

BUILDING BLOCKS



BALANCER

Nodo che autentica i client e ne gestisce le richieste redistribuendole equamente tra i vari edge.



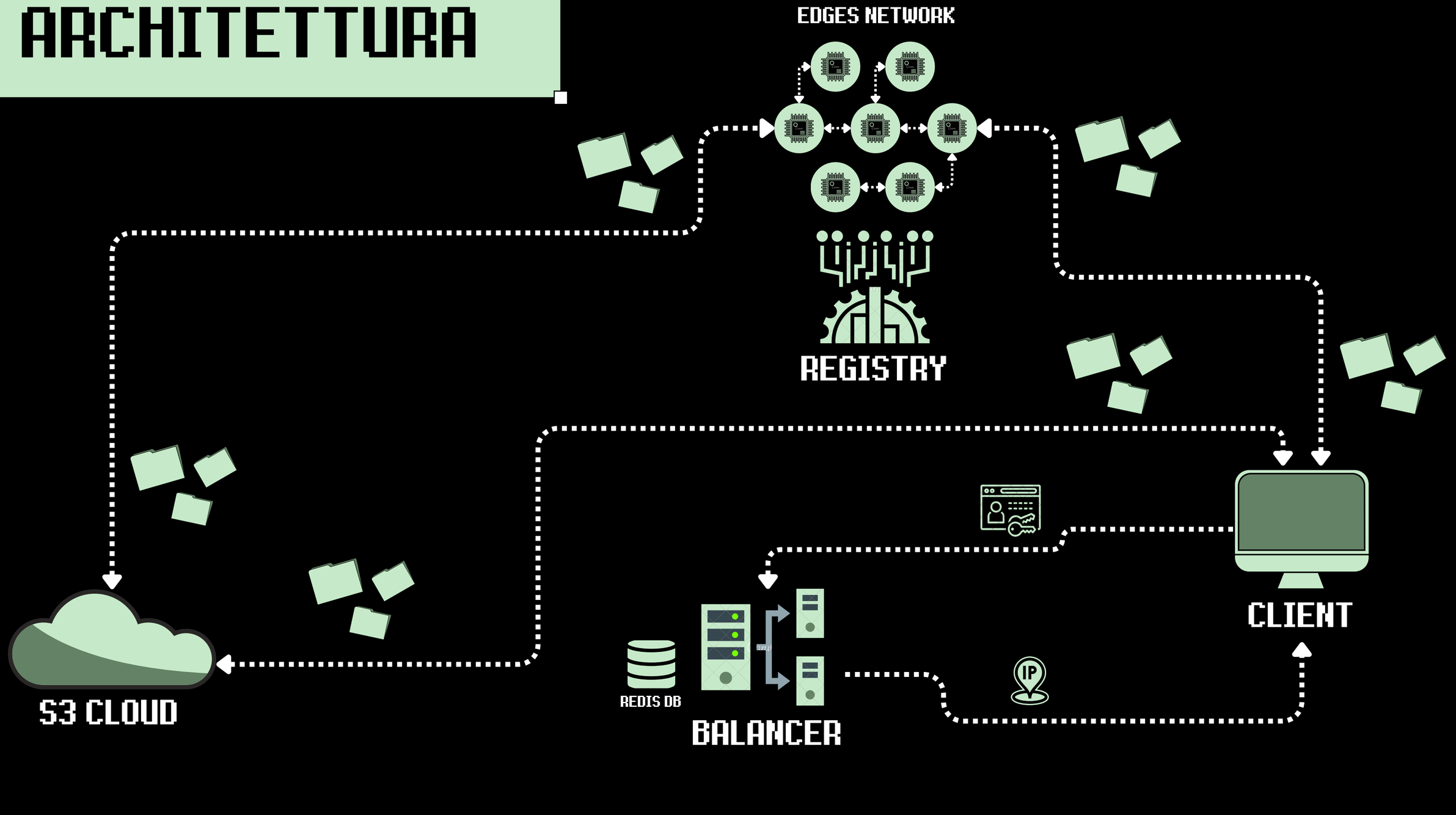
REGISTRY

Nodo che gestisce la rete di edge in maniera centralizzata con controlli periodici sulle componenti connesse.

REDIS DB

DB ospitato dal nodo del load balancer che permette di mantenere le utenze autorizzate in persistenza.

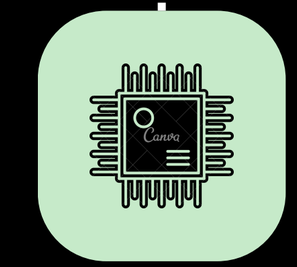
ARCHITETTURA



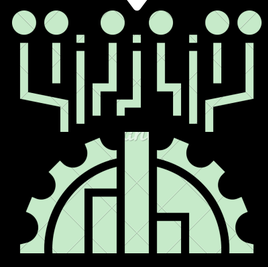
ENTRATA DI UN EDGE

Hello, i want to enter in the network!

Hey there! Here are your neighbours...



NEW EDGE !!!



REGISTRY

ELABORAZIONE

Scelta dei vicini a cui connettersi e aggiornamento degli edge attivi.

ELABORAZIONE

Tentativo di connessione ai vicini restituiti, inizio meccanismo di ping verso di essi e invio di heartbeat periodici verso il registry.

PING

TRA VICINI

27%

Connecting to neighbour... *OK!*

Connection request received, connecting... *OK!*

A FEW MOMENTS LATER...

Hey neighbour! Are you there?

Yes! Still here...

N SECONDS LATER...

Hey neighbour! Are you there?

Yes! Still here...

I am already connected to this guy... all good!

EDGE

NEIGHBOUR

PING

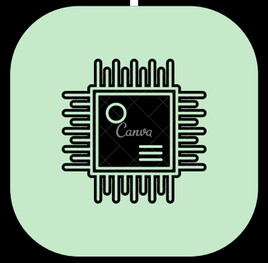
TRA VICINI

32%

Hey neighbour! Are you there?

Yes! Still here but... who are you?

I don’t know you...
Let’s connect!



Connection request sent...

OK!

EDGE

NEIGHBOUR

N SECONDS LATER...

Hey neighbour! Are you there?

Yes! Still here...

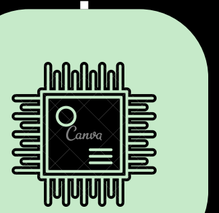
PING

TRA VICINI

Hey neighbour! Are you there?

I’m asking you for the 2nd time... Are you there?

Third time’s the charm... Are you there?



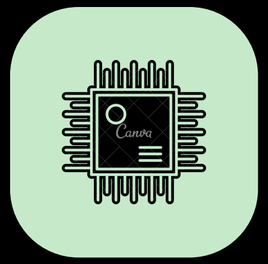
EDGE

NEIGHBOUR

Neighbour not responding...
assuming he’s dead...

HEARTBEAT

VERSO IL REGISTRY

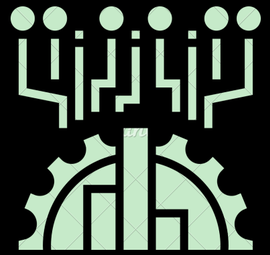


EDGE

I'm alive!

N SECONDS LATER...

I'm alive!

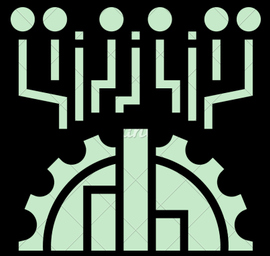


REGISTRY

Not hearing from edge for a while... assuming he's dead...

EDGE

T SECONDS LATER...

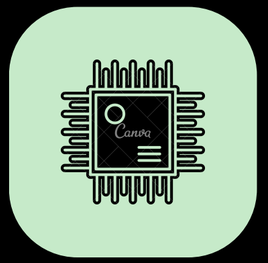


REGISTRY

COSTRUZIONE GRAFO

NEL REGISTRY

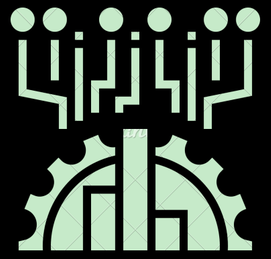
45%



EDGE

Which are your neighbours?

These are my neighbours!



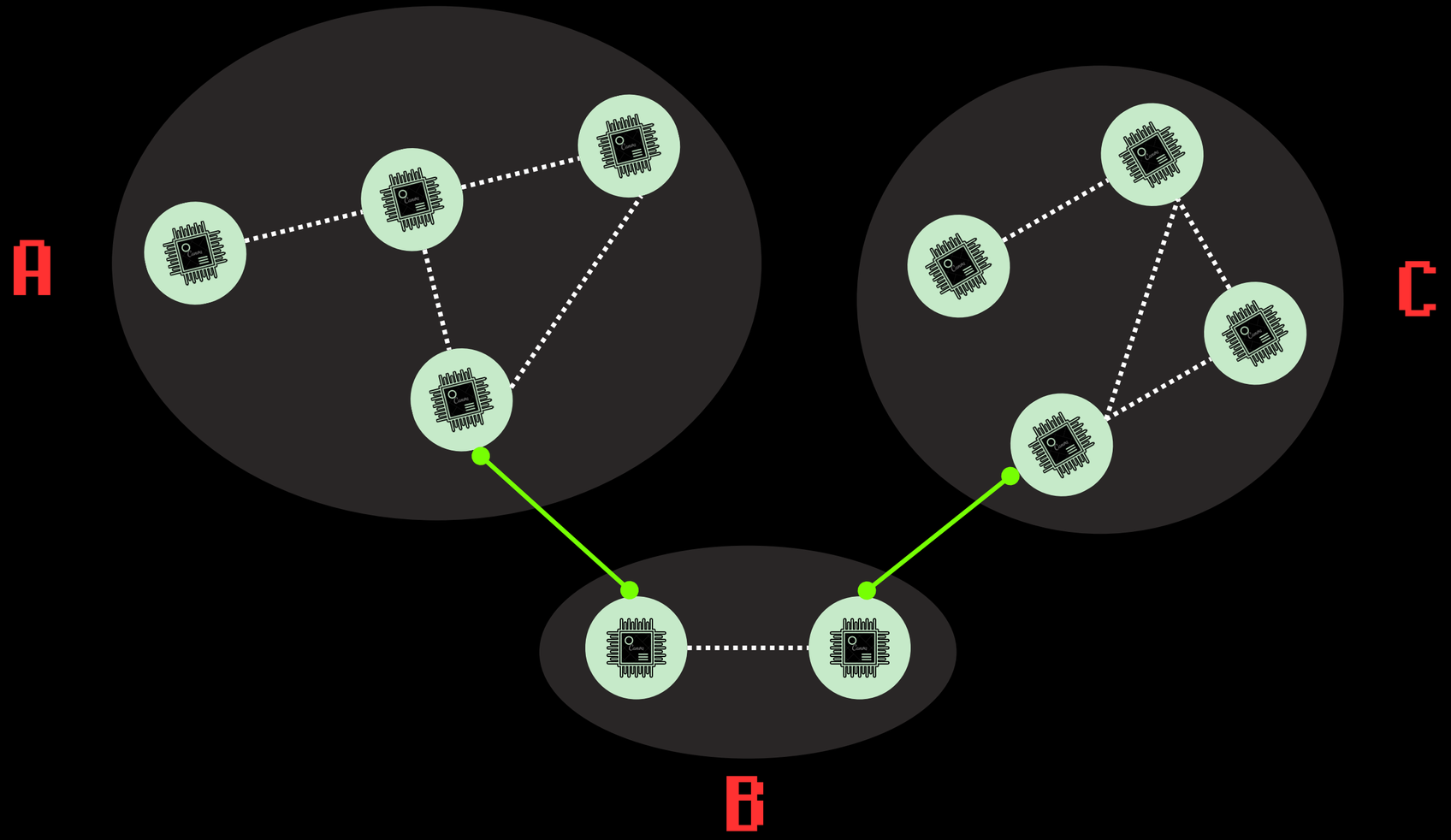
REGISTRY

ELABORAZIONE

Raccolgo le informazioni sui vicini per ciascun edge attivo e ricostruisco il grafo delle connessioni.

RICERCA COMPONENTI CONNESSE NEL REGISTRY

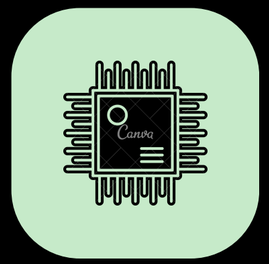
Una volta ricostruito il grafo delle connessioni...



HEARTBEAT

VERSO IL BALANCER

54%

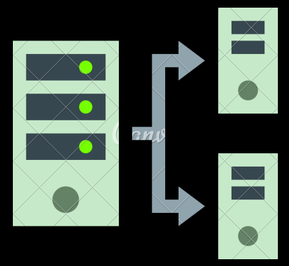


EDGE

I'm alive and i am serving X requests!

N SECONDS LATER...

I'm alive and i am serving Y requests!



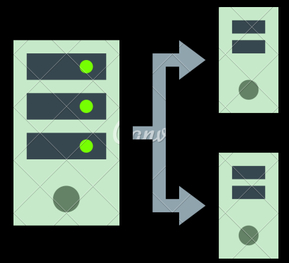
BALANCER

ELABORAZIONE

Se conosco l'edge aggiorno il *CURRENT_LOAD*, altrimenti lo aggiungo agli edge attivi e salvo il valore di carico.

EDGE

T SECONDS LATER...



BALANCER

Not hearing from edge for a while... assuming he's dead...

CONNESSIONE AL BALANCER

edge.DOWNLOAD/UPLOAD/DELETE “temp_sensorK.json”

username: sensorA
password: s3cr3t

Give me an Edge to contact.

CLIENT

BALANCER

Allow/Deny

REDIS DB

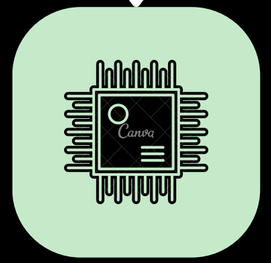
search for user

```
IF isLoggedIn(Client) {
    edge := ChooseEdgeWithMinLoad()
    edge.CURRENT_LOAD++
    return edge.IPaddr
}
```

DOWNLOAD REQUEST

64%

DOWNLOAD
“temp_sensorK.json”



EDGE

RICERCA DEL FILE

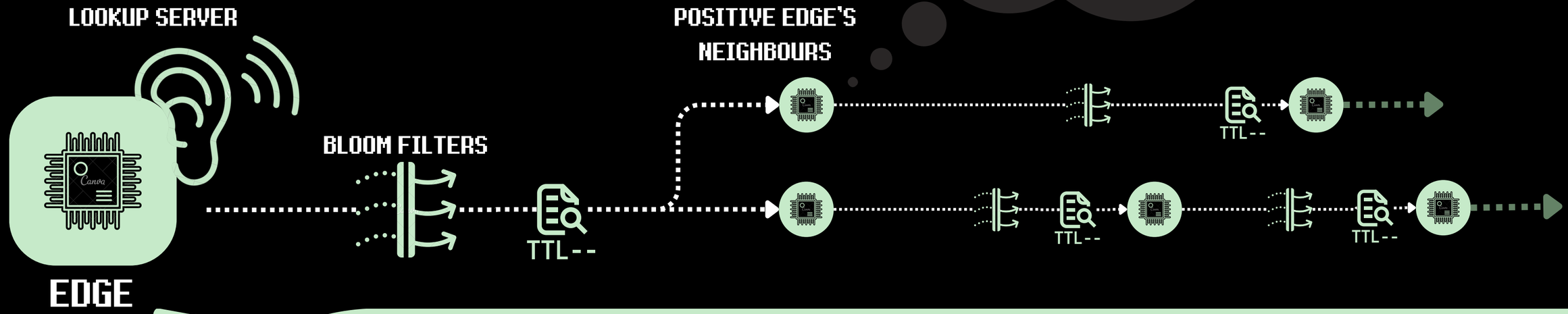
1. Ricerca del file nella cache locale
2. Ricerca del file nella rete
3. Ricerca del file su S3

LOOKUP DEL FILE

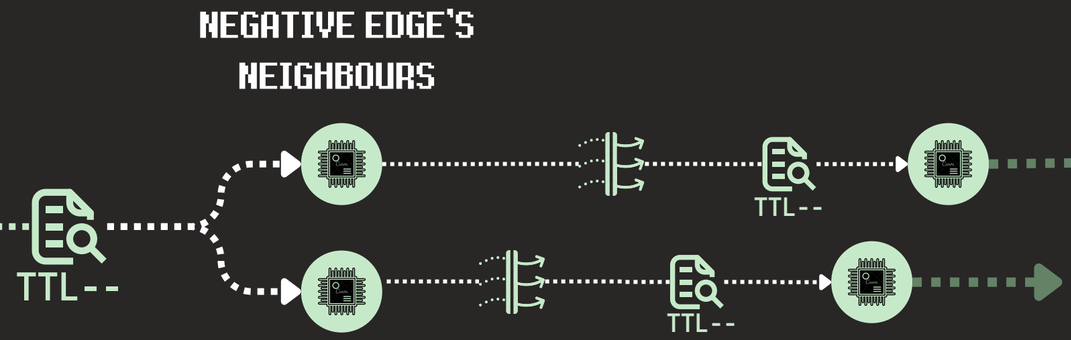
INVIO DELLE RICHIESTE

```
IF req NOT IN ReqCache {
    insertInReqCache(req)
} ELSE {
    ignoreReq(req)
}
```

68%



Not enough contacted?



LOOKUP DEL FILE

MECCANISMO DI CALLBACK

73%

I have it! Here is my ip...

FILE FOUND

LOOKUP SERVER

N SECONDS LATER...

Hey you! Give me the file...

EDGE

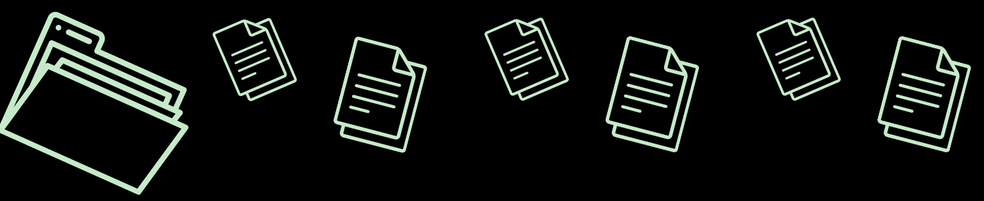
OWNER EDGE

FILE CHUNKS

UPLOAD REQUEST

77%

UPLOAD “temp_sensorK.json”

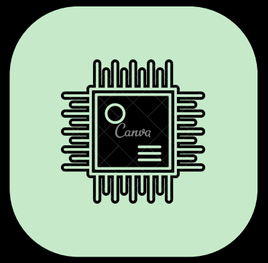


file chunks

+

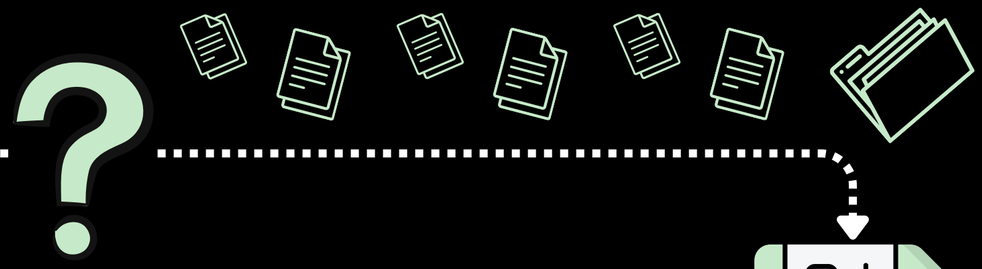
file

metadata



EDGE

S3 CLOUD



EDGE'S CACHE

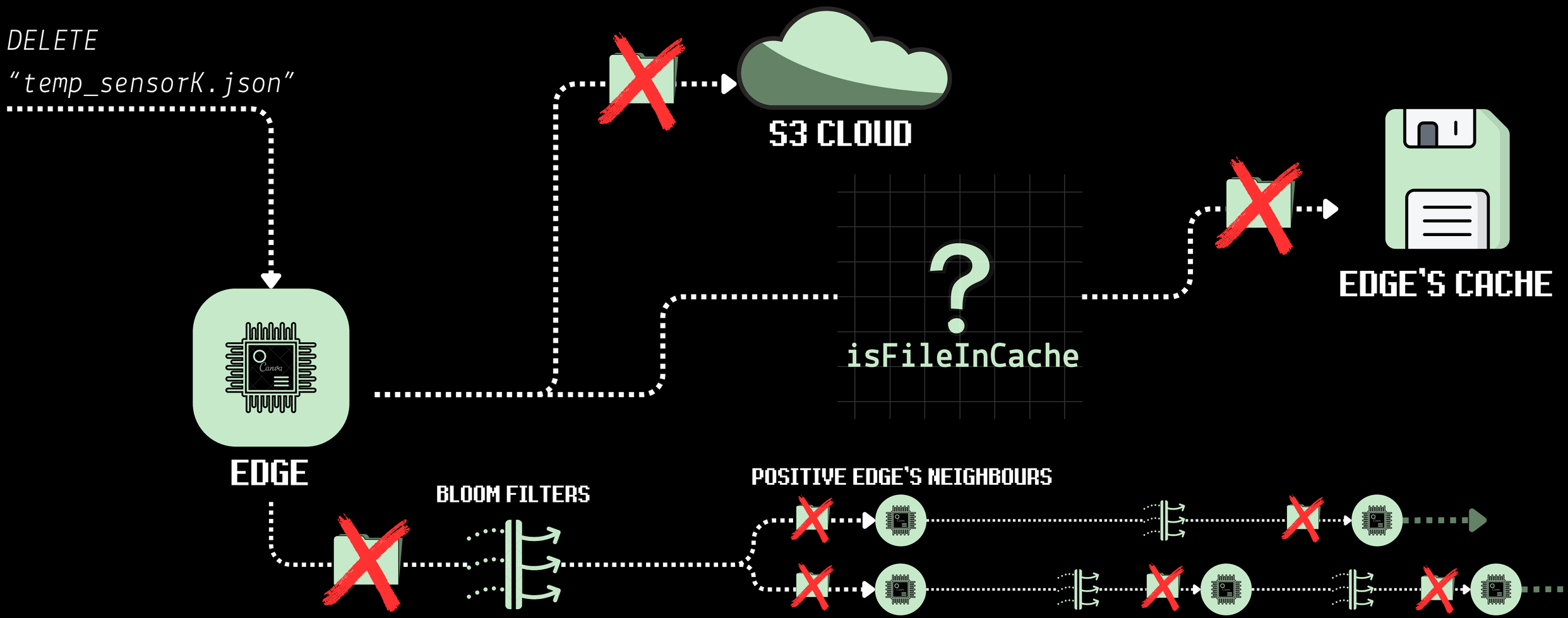
```
IF isFileCachable(fileSize) {
    cache.InsertFile(“temp_sensorK.json”)
}
```

DELETE REQUEST

82%

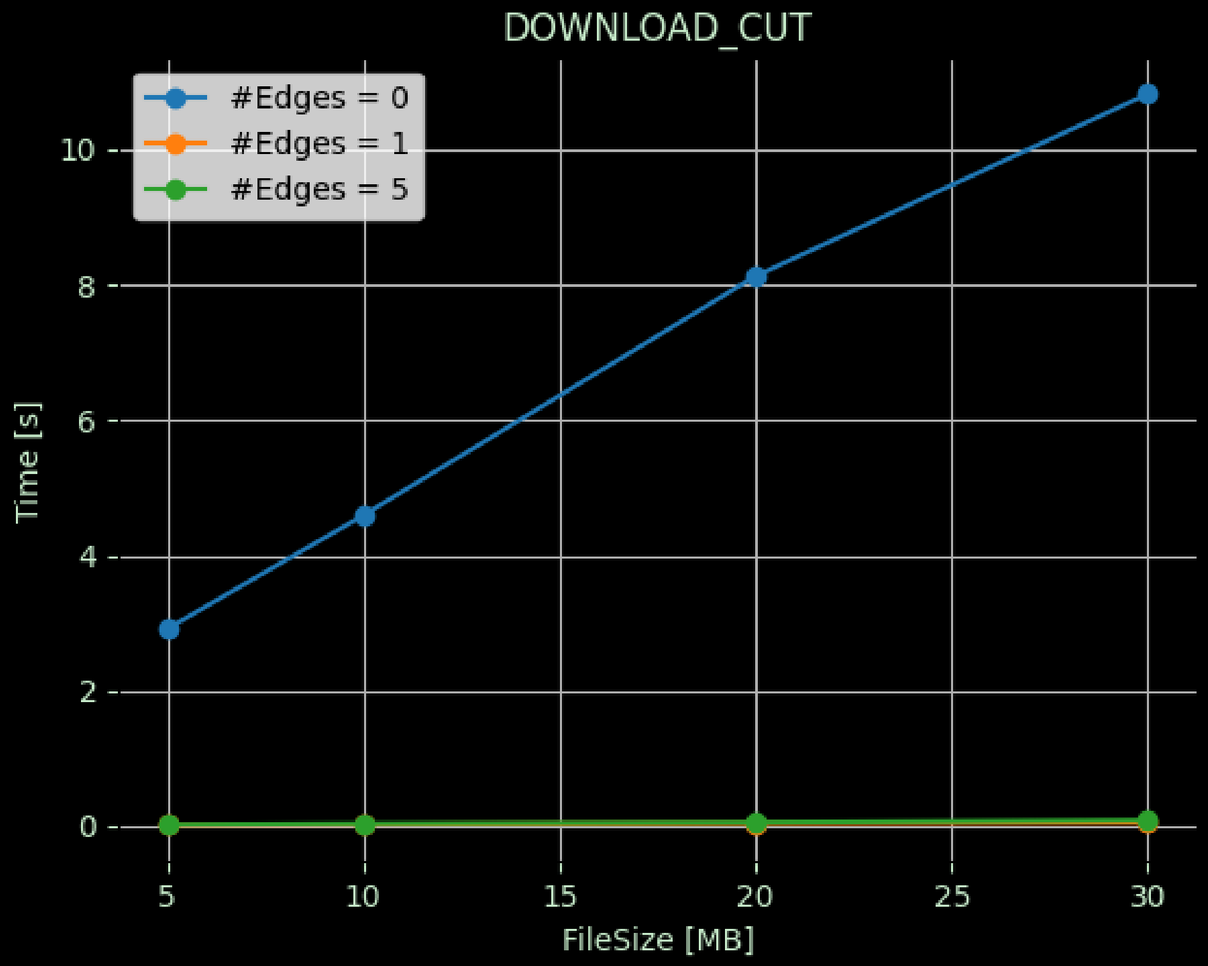
DELETE

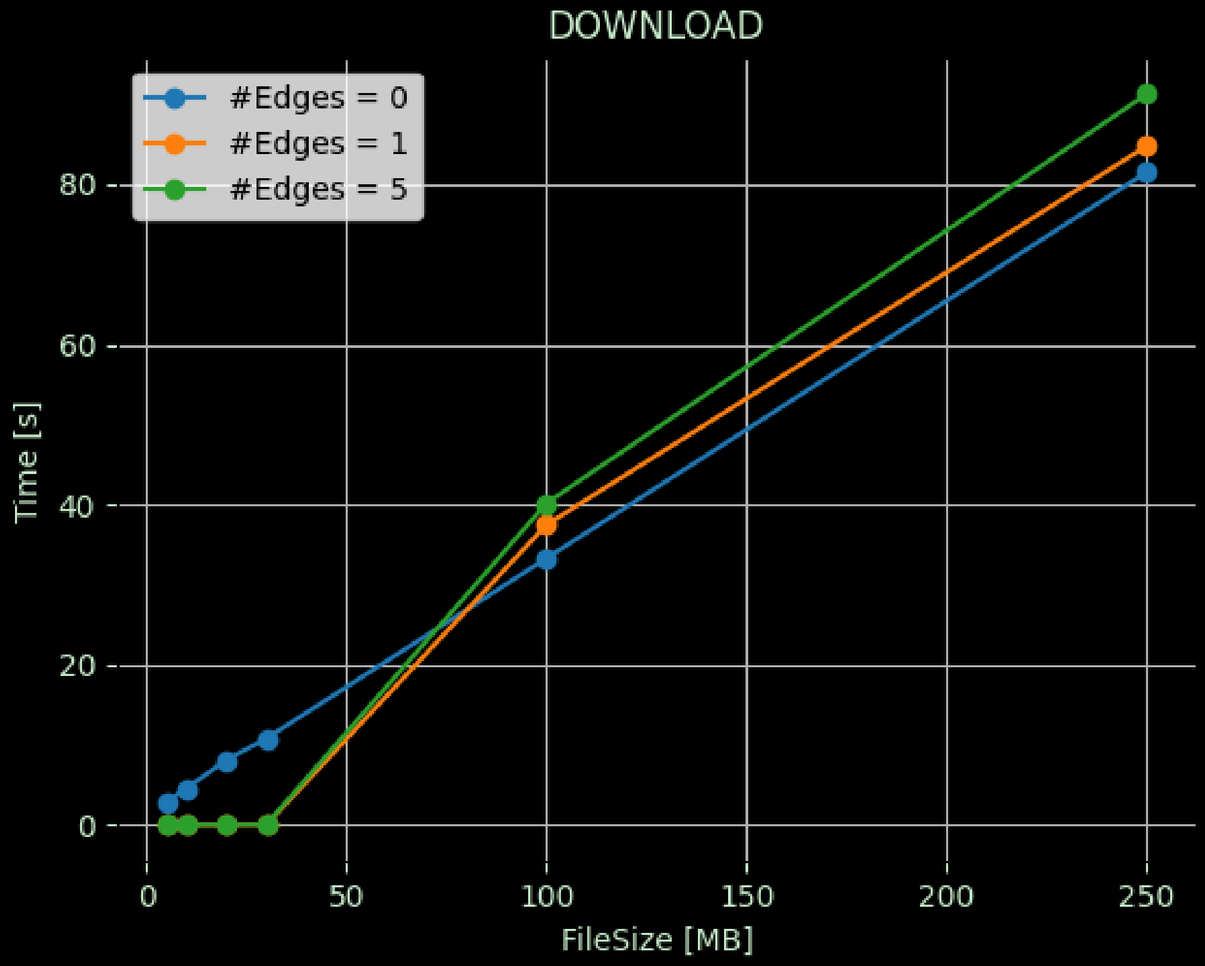
“temp_sensorK.json”



RISULTATI

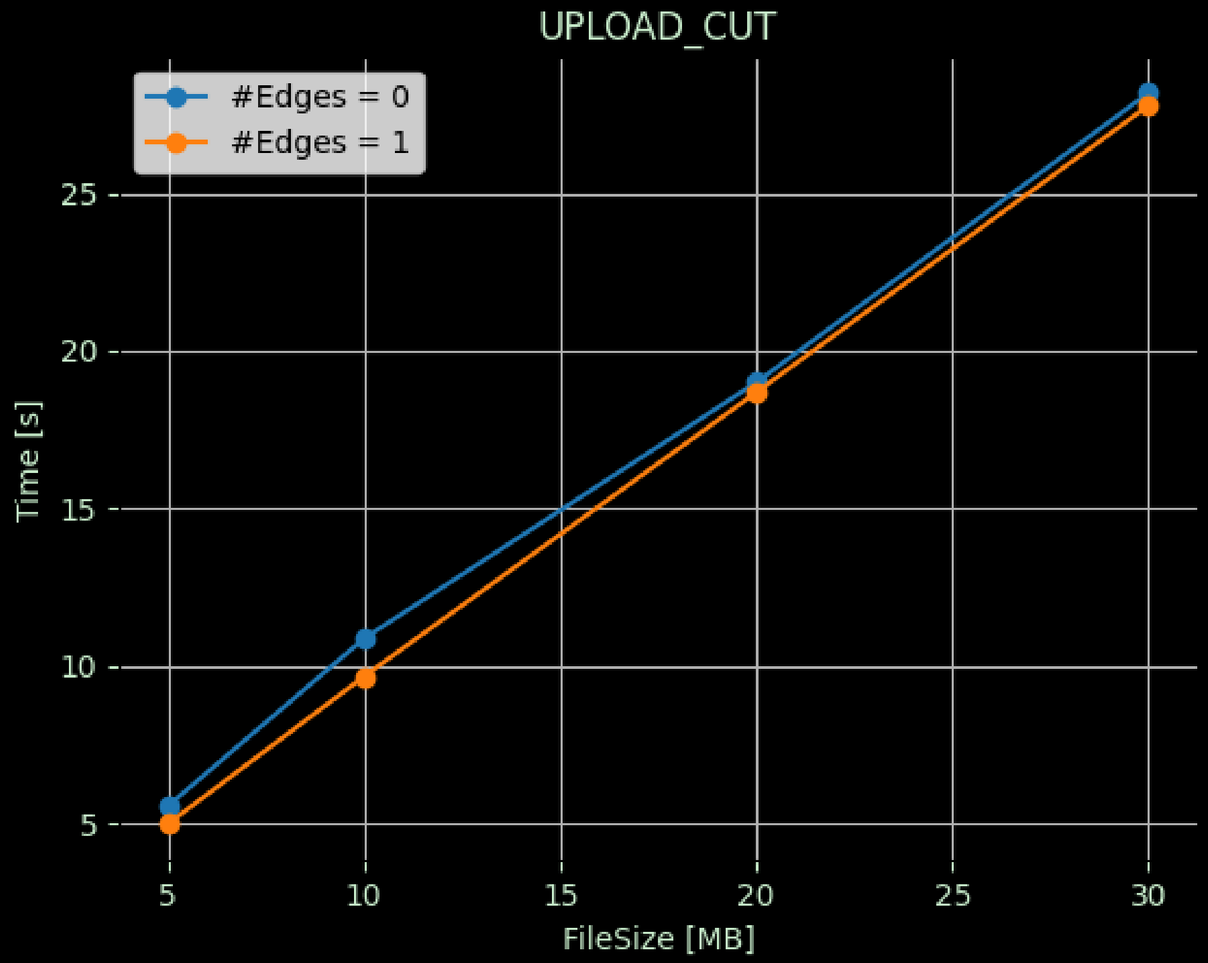
TEST SEQUENZIALI

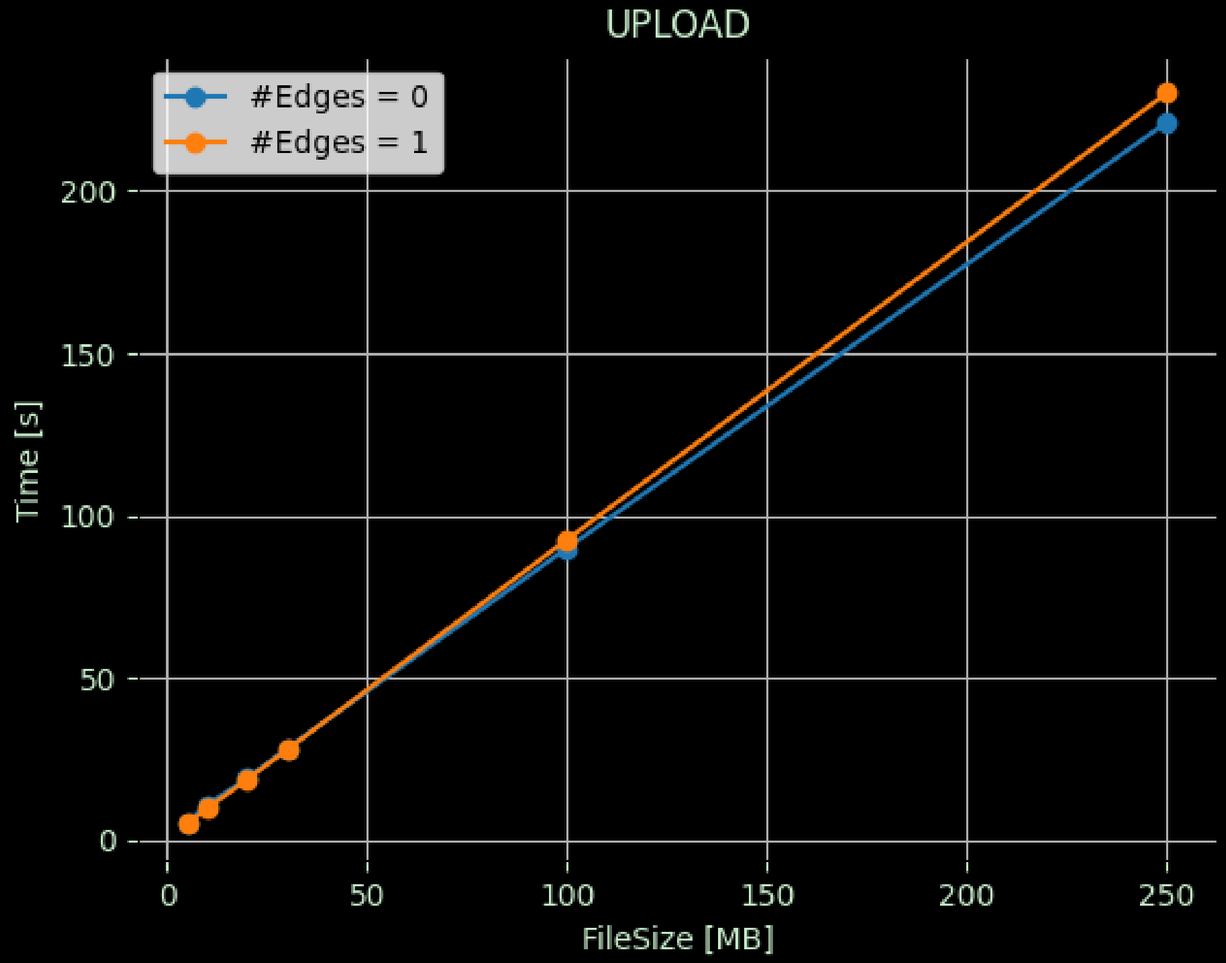




RISULTATI

TEST SEQUENZIALI

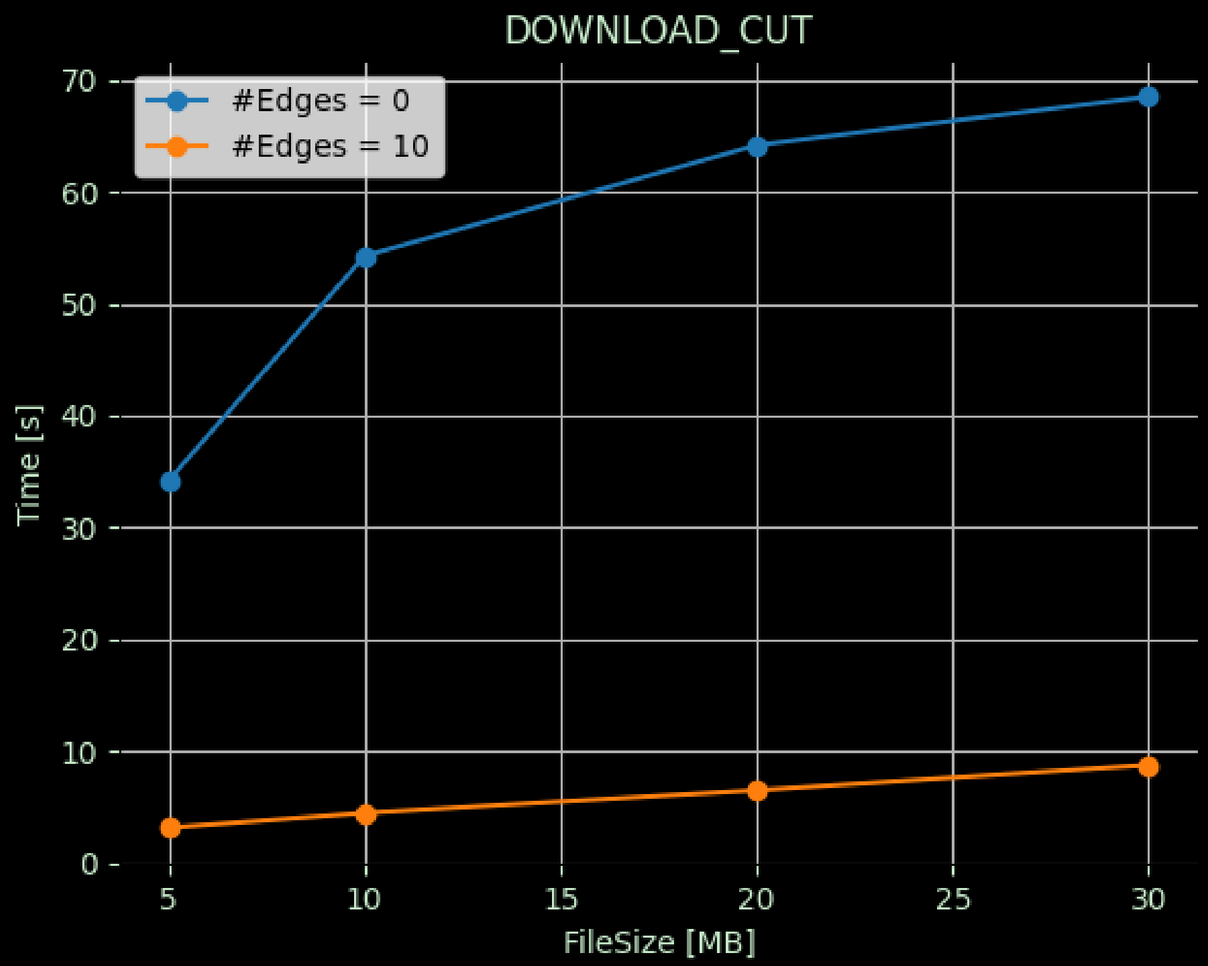


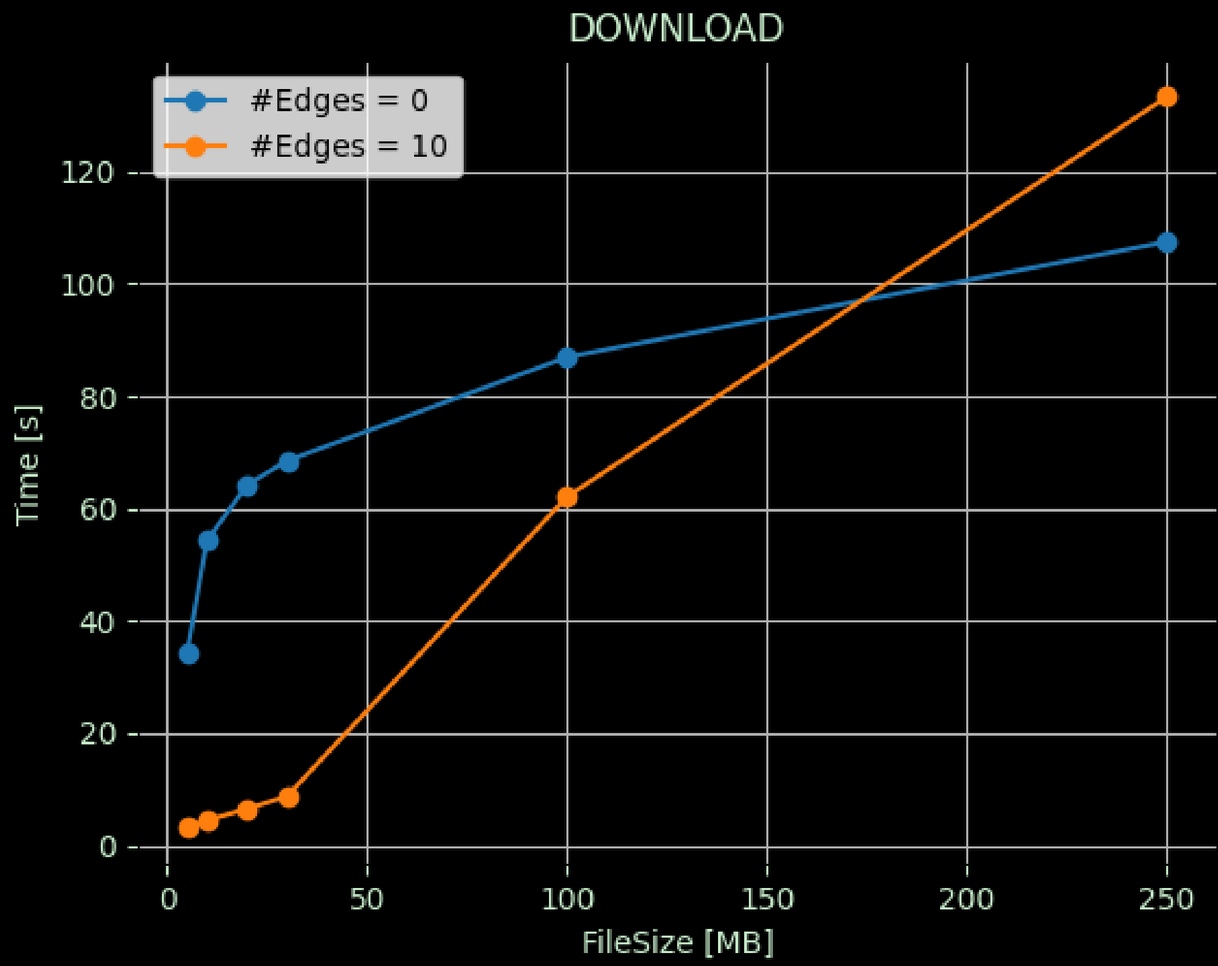


RISULTATI

TEST DI CARICO

95%





SVILUPPI FUTURI

- Supporto per il *versioning* sui file nel sistema.
- Miglioramento del meccanismo di recovery sui file scaricati da *Owner Edges* in maniera da supportare la ripresa dello scaricamento dal punto di interruzione.
- Analisi sugli effetti delle variazioni dei parametri configurabili per la ricerca di configurazioni ottimali del sistema *SAE Storage System* in rapporto al caso di applicazione.

Manenti Edoardo

Simone Nicosanti

Andrea De Filippis

GAME COMPLETED

A.A. 2022/2023

GRAZIE DELL'ATTENZIONE

Presentazione del progetto

"SAE storage service"

hope you liked it!