

# Sistema Client-Server Multithread con Autenticazioni per lo Streaming Video Multi-Utente

Antonio Santagati   Luigi Villari

Presentazione Progetto Laboratorio Reti e Sistemi Distribuiti

Anno Accademico 2023/2024

# Indice

- 1 Stato dell'arte
  - Streaming Video
  - Architettura client-server
  - Multithreading
  - VLC
- 2 Descrizione del Problema
- 3 Implementazione
  - Server
  - Verify
  - Manage log
  - Handle Client
  - Client
- 4 Risultati Sperimentali
- 5 Conclusione e Sviluppi Futuri

# Streaming Video

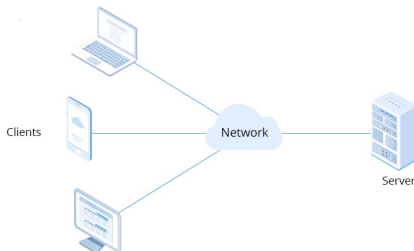
Lo streaming video è una tecnologia che consente la trasmissione continua di contenuti video attraverso una rete, senza dover scaricare interamente il file prima di poterlo visualizzare. Questa tecnologia permette agli utenti di guardare video in tempo reale, man mano che i dati vengono trasferiti dal server al dispositivo dell'utente.



# Architettura client-server

L'architettura client-server è un modello di progettazione di rete in cui le applicazioni sono suddivise in due componenti principali:

- 1 client: richiede e utilizza tali risorse o servizi.
- 2 server: fornisce risorse o servizi



# Multithreading

## Multithreading

Il multithreading è una tecnica di programmazione che permette l'esecuzione simultanea di più thread all'interno di un singolo processo, migliorando le prestazioni e la reattività del sistema.

# VLC

## VLC

VLC Media Player, comunemente noto come VLC, è un lettore multimediale open-source e cross-platform sviluppato dal progetto VideoLAN.

# Descrizione del Problema

Lo sviluppo di un sistema client-server multithread con autenticazione per lo streaming multi-utente rappresenta una sfida significativa. Il progetto prevede la realizzazione di un'infrastruttura in grado di gestire contemporaneamente più utenti e offrendo uno streaming video in tempo reale.

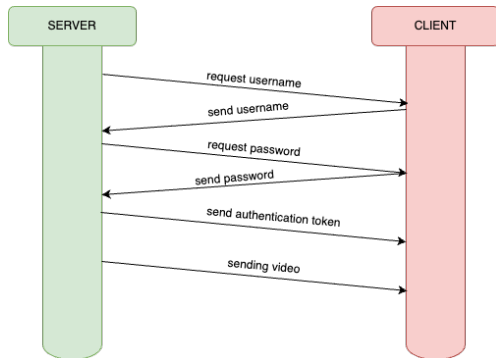
# Implementazione Server

- **Verify:** Autentica gli utenti.
- **Manage log** Verifica se l'utente ha già effettuato l'accesso.
- **Handle Client:** Gestisce la connessione con il client.



# Verify

La funzione viene richiamata nel momento in cui il server riceve le credenziali dal client. Verifica all'interno del file di testo se sono presenti, in caso contrario restituisce "Autenticazione negata".



## Manage log

Per evitare race conditions all'interno della funzione, è presente una sezione critica per sincronizzare i thread nell'utilizzo del file condiviso. Questo garantisce che solo un thread alla volta possa accedere alla risorsa condivisa, prevenendo accessi simultanei che potrebbero causare inconsistenze nei dati.

```
acquire(lock)
```

```
// Sezione Critica
```

```
// Scrittura o rimozione user dal file di log
```

```
release(lock)
```

```
// Sezione Non Critica
```

# Handle Client

La funzione gestisce la comunicazione con un client in un contesto di rete multithread. Riceve l'username e la password del client per autenticazione. In caso di successo, registra l'utente come connesso e invia una lista di video disponibili. Riceve la scelta del video dal client, apre il file corrispondente e lo invia in blocchi.

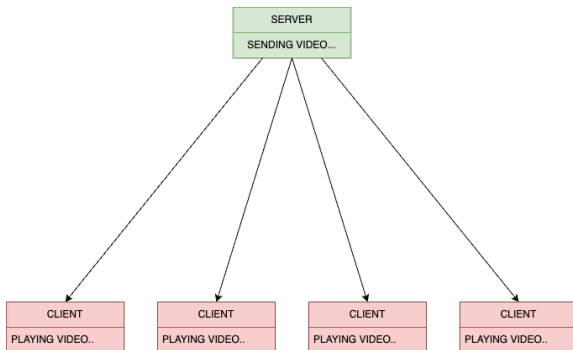
# Implementazione Client

- **init socket():** Configura e stabilisce la connessione con il server.
- **Autenticazione:** Gestione dell'invio di username e password per accedere ai video disponibili.
- **Selezione e riproduzione video:** Una volta autenticato, l'utente può selezionare un video da una lista fornita dal server e iniziare la riproduzione.

**Interazione Utente:** Il client offre un'interfaccia semplice per l'input dell'utente, permettendo di gestire pause, stop e riproduzione del video selezionato.

## Riproduzione Video

Per permettere la riproduzione video, su ogni client viene istanziato vlc in maniera autonoma, creando il media player e avviando la riproduzione in tempo reale tramite la lettura dei byte dalla socket.

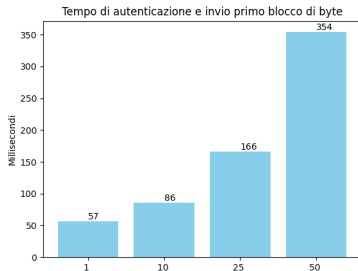


# Risultati Sperimentali

I risultati sperimentali sono stati ottenuti dopo molteplici prove effettuate in vari scenari per analizzare le prestazioni a massimo carico il sistema, analisi effettuate sulla robustezza dell'applicazione e sulla latenza e il throughput. I test a carico sono stati effettuati all'interno dell'open-lab dove 160 client simultaneamente aperti non hanno causato nessuna latenza per il server.

## Dati Riscontrati

Analizzando il tempo che intercorre tra l'autenticazione e l'invio del primo blocco di byte da parte del server è stato fondamentale per determinare che questo intervallo aumenta progressivamente all'aumentare del numero di client simultanei lanciati su un dispositivo. I principali test sono stati effettuati per 1 , 10 ,25 e 50 client.



## Conclusione e Sviluppi Futuri

In conclusione al progetto le nostre proposte per eventuali sviluppi futuri potrebbero essere l'aggiunta di una collezione più ampia di video, divisione in categorie e lo sviluppo di un'interfaccia utente facile e intuitiva. L'aggiunta di sistemi per l'interruzione momentanea di un video per riprenderlo in un secondo momento. Infine il progetto di sviluppo di un sistema client-server multithread per lo streaming video con autenticazione ha dimostrato come le moderne tecnologie di comunicazione e sicurezza possano essere integrate per offrire un servizio di streaming video affidabile e sicuro.



## referimenti

Tutta l'implementazione è ampiamente descritta e consultabile:  
<https://github.com/AntonioSantagati/PROGETTO-LAB>