POLITECNICO DI BARI

DEPARTMENT OF ELECTRICAL AND INFORMATION ENGINEERING

MASTER'S DEGREE IN COMPUTER ENGINEERING

Report Project

CONTROL METHODS FOR COMPUTER NETWORKS

Shaka-adv

Teachers:
Prof. Ing. Saverio Mascolo
Ing. Carlo Croce
Ing. Michele De Palma

Student:
Calia Angelo 590168

Academic Year 2022/2023

Riassunto

Il progetto in questione riguarda lo sviluppo di una pagina web che implementa shaka-player con un advertisement iniziale di 10 secondi. Realizzando infine un imagine docker.

Shaka-player fornisce un player video per la riproduzione del video e per il controllo, tra cui una barra di avanzamento del video, pulsanti di riproduzione/pausa, controllo del volume e del mute, nonché la possibilità di selezionare la qualità del video da riprodurre.

Per migliorare le prestazioni della riproduzione del video, viene inoltre implementato un meccanismo di caching che permette di caricare il video in modo più veloce e fluido, senza dover attendere il buffering del video stesso.

Inoltre, è stata integrata una pubblicità iniziale utilizzando il Google IMA. Questo ha permesso di inserire annunci pubblicitari all'inizio del video.

Una volta completata la pagina web, lo step successive è stato creare un'immagine Docker contenente tutto il progetto. Questo permette di creare facilmente un'istanza della pagina web su un server e di distribuire il progetto in modo semplice e portabile. L'immagine Docker è stata quindi testata per assicurarsi che funzioni correttamente e che la riproduzione del video avvenga senza problemi.

Il progetto sviluppato consente agli utenti di riprodurre video in streaming con pubblicità iniziale in modo semplice ed efficace, grazie all'utilizzo di shaka-player e del Google IMA. Inoltre, l'impiego di un'immagine Docker semplifica la distribuzione del progetto e ne garantisce la scalabilità.

Sommario

Shaka-Player

Shaka Player, è una libreria JavaScript open source sviluppata da Google per la riproduzione di video in streaming su browser e dispositivi mobili. La libreria è stata progettata per supportare lo streaming adattivo e il formato di trasmissione HTTP Live Streaming (HLS) su browser moderni, consentendo una riproduzione di video fluida e di alta qualità.
Shaka Player supporta anche il formato MPEG-DASH (Dynamic Adaptive Streaming over HTTP), che è un formato di streaming adattivo per la trasmissione di contenuti audio e video su Internet. Ciò significa che Shaka Player adatta automaticamente la qualità del video in base alla larghezza di banda disponibile del dispositivo, fornendo un'esperienza di visualizzazione ottimale per gli utenti.
Shaka Player è altamente personalizzabile, consentendo agli sviluppatori di integrare la libreria all'interno di qualsiasi applicazione web e di configurare il comportamento della riproduzione video in base alle esigenze specifiche.
Per ottenere il codice sorgente e successivamente compilare la libreria, sono necessari dei prerequisiti:

- Git versione 1.9+
- Python versione 2.7 o 3.5+
- Java Runtime Environment versione 14+
- NodeJS versione 14+

Una volta terminata l’installazione dei prerequisiti si può scaricare il codice da git con il seguente comando

```
git clone https://github.com/shaka-project/shaka-player.git
cd shaka-player
```

Infine, compilare la libreria per generare il file necessari tramite

```
python build/all.py
```

Google IMA SDK

Shaka Player fornisce un'API per la pubblicazione di annunci. L’ API è stata progettata per integrazione con i Software Development Kit (SDK) di Interactive Media Ads (IMA), in modo tale da consentire la visualizzazione di annunci all'interno di video online.
Un SDK è un insieme di strumenti software che gli sviluppatori possono utilizzare per creare applicazioni o funzionalità specifiche all'interno di un'applicazione esistente. Interactive Media Ads (IMA) è una suite di SDK che semplifica l'integrazione di annunci multimediali nei siti web e nelle app. Gli SDK IMA possono richiedere annunci da qualsiasi ad server conforme a VAST e gestire la riproduzione degli annunci.
Esistono due modi per implementare gli SDK IMA: lato client, che combina video di annunci e contenuti all'interno dell'app, e inserimento dinamico di annunci (DAI), che combina video di annunci e contenuti nei server Ad Manager; quindi, restituisce un singolo flusso video all'app per la riproduzione. Nel nostro progetto verrà utilizzato im metodo lato client.

Codice

La scrittura del codice prevede tre file principale:

- index.html
- style.css
- myapp.js

Nel tag <head> del file html vengono richiamati i vari script necessari per Shaka, l'advertisement fornito da Google e infine lo stile del file css.
Nel tag <body> viene creato il contenitore per Shaka e quello per la sorgente video.
Nel file myapp.js avviene impostato il video di riferimento e salvato nella constante 'manifestUri ', l'URL è suggerito direttamente nella documentazione di Shaka-player.

Il codice richiede e avvia la riproduzione di annunci video all'interno di un lettore multimediale utilizzando le librerie Google IMA e Shaka Player. Inoltre, specifica i dettagli della richiesta di pubblicità, invoca il metodo per richiedere l'annuncio e ascolta gli eventi per gestire la riproduzione degli annunci.

Nella riga 16 viene impostato URL dell'advertisement fornito da Google IMA come prova. Tramite la riga 35 viene caricato un video e gestito eventuali errori che si verificano durante il caricamento. Se il caricamento ha successo, viene stampato un messaggio di log sulla console.

Docker

Docker è una piattaforma di virtualizzazione a livello di sistema operativo che consente di creare e gestire container, ovvero ambienti isolati e autonomi in cui eseguire le applicazioni. Grazie all'uso dei container, Docker offre numerosi vantaggi, come la possibilità di confezionare un'applicazione insieme a tutte le sue dipendenze e configurazioni in un pacchetto portatile chiamato "immagine Docker".
L'immagine Docker può essere eseguita su qualsiasi sistema operativo che supporta Docker, senza dover preoccuparsi di eventuali incompatibilità o conflitti di versione tra le librerie o le dipendenze. Ciò rende l'esecuzione dell'applicazione molto più semplice e portabile, eliminando la necessità di installare manualmente tutte le dipendenze.
Per creare un'immagine Docker, si utilizza un file di testo chiamato Dockerfile.
Il Dockerfile contiene le istruzioni per definire l'ambiente di esecuzione dell'applicazione, inclusi i pacchetti e le librerie necessarie, le variabili d'ambiente e le porte da esporre.
Una volta definito il Dockerfile, è possibile utilizzarlo per creare l'immagine Docker, che può essere eseguita in un container Docker. Il container è un'istanza dell'immagine Docker, che consente di eseguire l'applicazione in un ambiente isolato e autonomo, senza interferire con altre applicazioni o risorse del sistema.

Dockerfile

Nel caso del progetto in questione, il Dockerfile utilizza come base l'immagine ufficiale di Nginx, "nginx:1.22.1", che è una versione stabile e leggera del server web ad alte prestazioni.

La scelta di questa immagine come base per l'immagine Docker del progetto è stata fatta per diversi motivi. In primo luogo, Nginx è uno dei server web più diffusi al mondo inoltre, l'immagine "nginx:1.22.1" è una versione stabile e aggiornata di Nginx, che garantisce una maggiore sicurezza e prestazioni migliori rispetto alle versioni precedenti.

Successivamente, nel Dockerfile viene effettuata l'installazione delle dipendenze necessarie per Shaka Player tramite il comando "RUN".

Una volta installate le dipendenze, viene impostata la directory di lavoro all'interno dell'immagine Docker in "\app" utilizzando il comando "WORKDIR". Questa operazione consente di specificare la directory di lavoro predefinita all'interno dell'immagine Docker, semplificando così la gestione dei file all'interno dell'immagine.

Infine, i file del progetto corrente vengono copiati nella directory "/usr/share/nginx/html/" di Docker utilizzando il comando "COPY". Questo consente di inserire i file necessari per l'applicazione all'interno dell'immagine Docker, rendendo l'applicazione completamente portatile e indipendente dall'ambiente di esecuzione.

Per concludere, con il comando "EXPOSE" viene esposta la porta 80 per il server web Nginx, consentendo così ai client di accedere all'applicazione tramite il protocollo HTTP.

Build

La build di un Dockerfile è il processo di creazione di un'immagine Docker a partire dalle istruzioni contenute nel file Dockerfile. In altre parole, la build di un Dockerfile è il processo di trasformazione di un insieme di istruzioni in un'immagine Docker completa, che contiene tutto ciò di cui un'applicazione ha bisogno per funzionare.

La build di un Dockerfile avviene tramite il comando "docker build", che viene eseguito da linea di comando all'interno della directory contenente il file Dockerfile. Il comando "docker build" esegue le istruzioni contenute nel Dockerfile in ordine sequenziale, creando una serie di layer che compongono l'immagine finale.

Una volta completata la build del Dockerfile, viene creata un'immagine Docker completa, che può essere utilizzata per eseguire l'applicazione in un ambiente Docker. L'immagine Docker è completamente portatile e indipendente dall'ambiente di esecuzione.

Nel caso specifico del progetto la build è stata fatta tramite il seguente comando nel terminale

```
docker build -t immagine_progetto .
```

il terminale ha restituito

```
[+] Building 146.2s (12/12) FINISHED
```

Dimostrazione del fatto che la build si è conclusa con successo.

Run

Il comando "docker run" avvia un nuovo container partendo da un'immagine Docker esistente e ne avvia l'esecuzione. Quando si esegue il comando "docker run", si specifica l'immagine Docker da cui creare il container e opzionalmente si specificano ulteriori opzioni come il mapping delle porte, il collegamento di volumi esterni, la configurazione delle variabili d'ambiente e così via.
Nel caso specifico del nostro progetto per avviare il container basta eseguire il seguente comando tramite il terminale

```
docker run -d -p 32667:80 --name container_progetto
immagine_progetto
```

L'output del terminale è ID univoco del container appena creato. È possibile utilizzare questo ID per accedere alle informazioni e alle operazioni del container tramite altri comandi Docker, ad esempio "docker ps" per visualizzare i container in esecuzione o "docker stop" per interrompere l'esecuzione del container.

Push

Il comando "docker push" consente di caricare un'immagine Docker su un registry, come ad esempio Docker Hub o un registry privato. Questo comando è utile quando si vuole condividere un'immagine Docker con altre persone o quando si vuole utilizzare l'immagine su un'altra macchina.

Per utilizzare il comando "docker push", è necessario avere un account sul registry di destinazione e aver eseguito il login utilizzando il comando "docker login". Una volta effettuato il login, è possibile caricare l'immagine Docker utilizzando il comando "docker push" seguito dal nome dell'immagine e dal tag.
Nel nostro caso è stato utilizzato come registry Docker Hub e effettuato il push tramite il seguente comando

```
docker tag immagine_progetto angelocal/progetto
```

Conclusioni e Sviluppi Futuri

L'obiettivo è stato raggiunto: l'applicazione rispetta i requisiti stabiliti.
Come primo sviluppo future, si è pensato di aggiungere un pulsante "skip ad" con un timer decrementale può essere una soluzione utile per migliorare l'esperienza degli utenti che utilizzano una piattaforma di video streaming. In questo modo, gli utenti potrebbero scegliere di saltare una pubblicità dopo un certo periodo di tempo, anziché essere costretti a guardare l'intera pubblicità.

Il secondo obiettivo è l'implementazione di varie pubblicità. Utilizzando diversi tipi di pubblicità, come ad esempio video pre-roll, mid-roll e post-roll, si potrebbe offrire una maggiore varietà di contenuti pubblicitari.

Infine, migliorare l'interfaccia grafica di Shaka player potrebbe essere una scelta importante per migliorare l'esperienza complessiva degli utenti. Una migliore interfaccia grafica potrebbe rendere la piattaforma più intuitiva e facile da utilizzare, aumentando così l'interesse degli utenti e migliorando la loro soddisfazione. Ciò potrebbe portare ad un aumento della fidelizzazione degli utenti, con un conseguente aumento del tempo trascorso sulla piattaforma e dei ricavi pubblicitari generati.