Emulazione di acceleratori hardware

Un approccio basato su QEMU

Laurea triennale in Informatica

Leonardo Ganzaroli (1961846)

A.A. 2024/25

SAPIENZA
UNIVERSITÀ DI ROMA

Indice

Cosa sono gli acceleratori hardware?

Componenti (hardware) progettati per eseguire operazioni specifiche in modo più efficiente rispetto ad una CPU tradizionale.

Schede grafiche, schede audio, coprocessori, ecc…

Un problema non trascurabile

I tempi di sviluppo complessivi sono importanti, solamente per l'hardware ci vogliono mesi.

Inoltre lo sviluppo della parte software non può davvero iniziare senza avere un prototipo fisico.

Cos’è un emulatore?

Un programma che riproduce il funzionamento di un altro PC **totalmente via software**.

Obiettivo del tirocinio

1 Introduzione

L’emulazione è uno strumento efficace per mitigare i problemi dello sviluppo software?

Come si cerca la risposta?

1. Creazione di un acceleratore hardware virtuale
2. Esecuzione di test sullo stesso

Indice

2 Elementi fondamentali

Un po’ di teoria

2 Elementi fondamentali

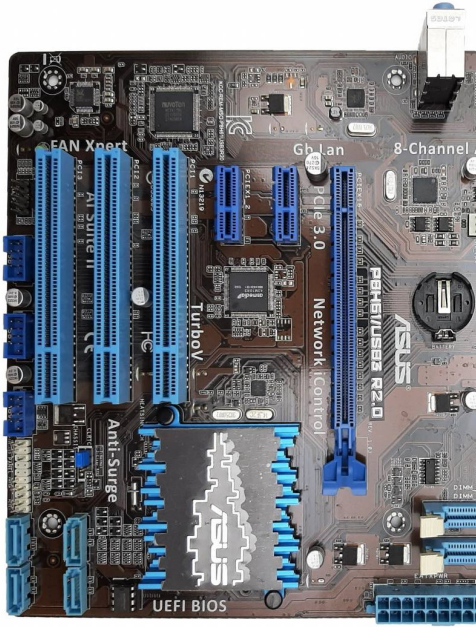
Prima di creare il prototipo ed emularlo è stato necessario approfondire 3 elementi fondamentali:

1. Metodi di collegamento degli acceleratori
2. Funzionamento dei driver (in Linux)
3. QEMU

PCIe

Peripheral Component Interconnect Express

- Metodo di collegamento largamente diffuso
- Molto complesso ma concettualmente funziona come una rete a commutazione di pacchetto
- Definisce come il sistema operativo deve interfacciarsi con i dispositivi e viceversa



Driver in Linux

2 Elementi fondamentali

Implementati tramite i **moduli**, componenti software che forniscono funzionalità aggiuntive al kernel.

A grandi linee gli elementi fondamentali di un driver sono:

- Array degli ID dei dispositivi supportati
- Funzioni per la gestione sia del modulo che dei dispositivi
- `struct` contenente le informazioni del driver
- Alcuni metadati

Associazione col dispositivo

L’associazione di un dispositivo PCIe al giusto driver avviene grazie alle informazioni che lo stesso fornisce al sistema operativo, tra queste ci sono gli stessi ID presenti nel driver.

QEMU

2 Elementi fondamentali

- Un emulatore generico
- Emula i dispositivi esterni al PC con un paradigma ad oggetti:
  - La classe equivale ad un file C presente nel codice sorgente
  - L'istanza viene creata basandosi sul file e immessa nel sistema emulato
- Con il linguaggio C e QOM[1] si possono creare dispositivi da o

[1] QEMU Object Model, un modello a cui attenersi.

Indice

3 Prototipo

Sistema emulato

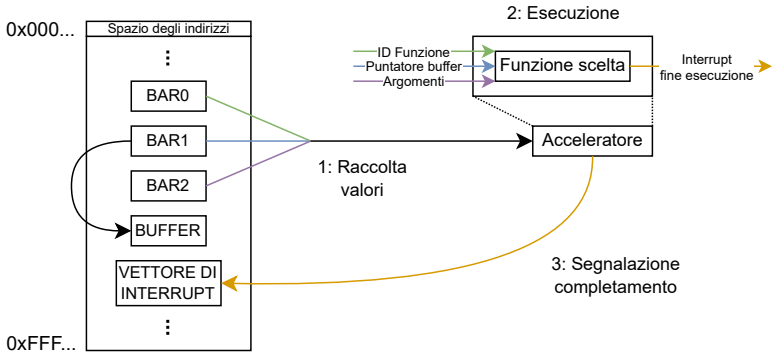
3 Prototipo

L'effettivo sistema usato è formato da:

- PC generico con architettura x86_64
- Kernel Linux
- Filesystem minimale

Acceleratore

3 Prototipo



Struttura + esempio di funzionamento

Driver

3 Prototipo

Composto essenzialmente da 3 parti:

1. Gestione del driver stesso
2. Gestione dell’acceleratore
3. Operazioni rivolte all’utente

Libreria

3 Prototipo

Astrae la comunicazione con il Driver, in breve permette all'utente di interagire "direttamente" con l'acceleratore.

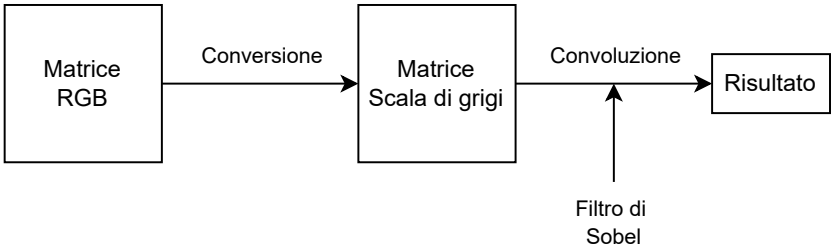
Per l'utente quindi risulta:

$$\text{funzione } x \text{ in libreria} = \text{funzione } x \text{ in acceleratore}$$

Indice

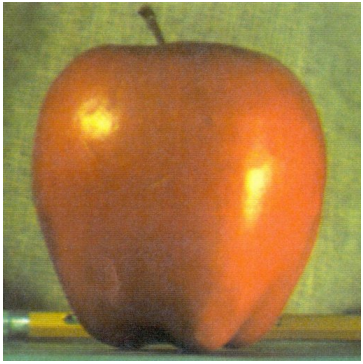
Operazione scelta

4 Test

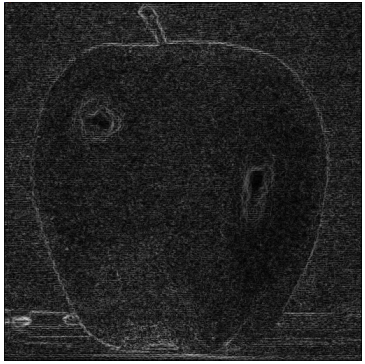


Esempio

4 Test



Input



Output

Confronto con un altro metodo

4 Test

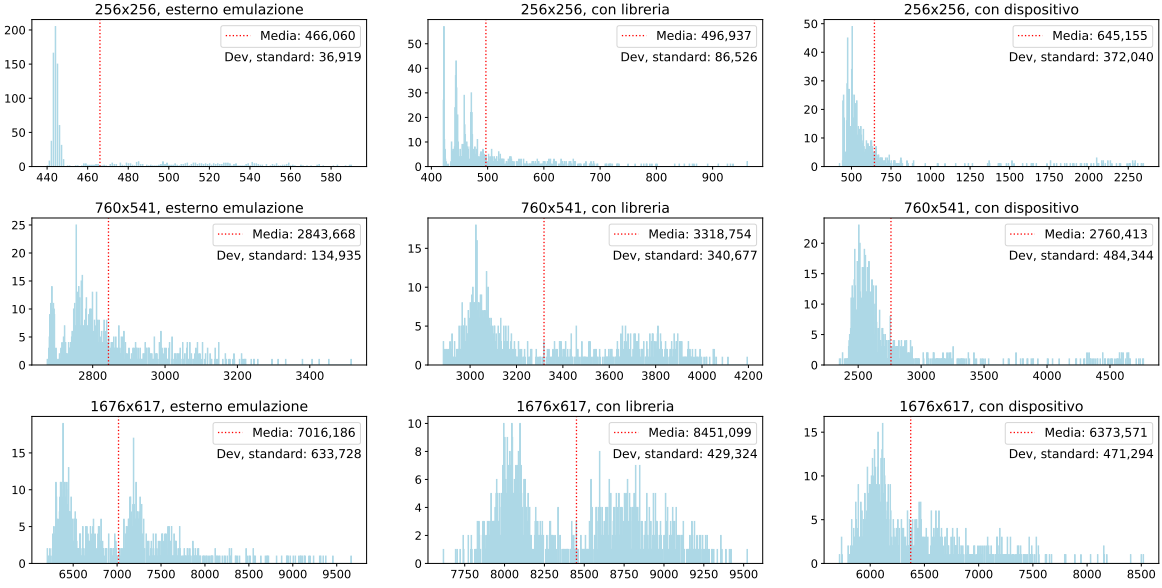
Il metodo più spartano per “emulare” un acceleratore è una semplice libreria.

I test hanno incluso anche questo metodo, risulta così possibile confrontare i 2 approcci.

Raccolta dei dati

4 Test

- Operazione svolta in 3 modi:
  - Con acceleratore
  - Con libreria fuori dall'emulazione
  - Con libreria nell'emulazione
- 3 matrici di grandezza differente $\rightarrow$ 9 casi totali
- 1000 misure per ogni singolo caso $\rightarrow$ 9000 misure totali



Istogrammi dei tempi rilevati (μS)

In percentuale

4 Test

Metodo	Piccola	Media	Grande
Lib. interna	+6,6%	+20,2%	+32,5%
Lib. esterna	-	+3%	+10,1%
Acceleratore	+38,4%	-	-

Indice

Limitazioni riscontrate

La natura stessa dell'emulazione (hardware "finto") non la rende adatta a contesti dove è necessario un certo grado di realismo, per esempio nel caso real-time.

QEMU non implementa tutte le funzionalità PCIe, alcune proprio per la mancanza di hardware.

Un ulteriore problema

5 Conclusioni

Commenti nel codice sorgente di QEMU

Conclusioni

L’emulazione, e in particolare QEMU, sono degli strumenti più che validi in questo ambito.

Nonostante le carenze permettono di creare ed emulare acceleratori hardware in modo abbastanza fedele e con buone prestazioni.

Fine.

Domande?