

Corso di Laurea in Ingegneria e Scienze Informatiche

B-Tree File System per piattaforme IoT

Tesi di laurea in:
SUPERVISOR'S COURSE NAME

Relatore

Marco Antonio Boschetti

Candidato

Leonardo Marcaccio

Abstract

Questa Tesi esplora il processo di studio, progettazione e implementazione di un prototipo di File System basato sulla struttura dati B-Tree, sviluppato per la piattaforma IoT IOtto di Onit S.p.A.

L'obiettivo principale è migliorare l'efficienza nella gestione e nel recupero dei dati, affrontando le problematiche tipiche di un File System quali la scalabilità e l'ottimizzazione delle risorse.

Il lavoro comprende un'analisi dello stato dell'arte sui moderni File System, con particolare attenzione alla loro interazione con i sistemi IoT. Viene approfondito il principio di funzionamento del B-Tree, dimostrando come questa struttura dati possa essere sfruttata per realizzare una struttura performante e affidabile.

Inoltre, vengono descritte le fasi di progettazione e implementazione, evidenziando le soluzioni adottate per adattare il prototipo alle esigenze specifiche della piattaforma IOtto.

I risultati preliminari mostrano che il File System proposto garantisce significativi miglioramenti in termini di velocità di accesso ai dati e utilizzo delle risorse rispetto alle alternative tradizionali.

Optional. Max a few lines.

Contents

Abstract	iii
1 Introduction	1
1.1 Background	1
1.1.1 Onit	1
1.1.2 IOtto	1
1.2 Descrizione del problema	1
1.3 Obiettivi di Tesi	1
1.3.1 Domande	2
1.3.2 Scopo	2
1.4 Struttura della Tesi	3
2 State of the art	5
2.1 Onit e IOtto	5
2.2 File System nell'IoT	5
2.2.1 Eterogeneità	6
2.3 B-Tree	6
2.4 B-Tree nella realizzazione dei File System	6
2.5 Some cool topic	6
3 Contribution	7
3.1 Fancy formulas here	7
	9
Bibliography	9

CONTENTS

List of Figures

LIST OF FIGURES

List of Listings

listings/HelloWorld.java	7
------------------------------------	---

LIST OF LISTINGS

Chapter 1

Introduction

1.1 Background

1.1.1 Onit

1.1.2 IOtto

AGGIUNGI PRESENTAZIONE DELL'AZIENDA E DEL PRODOTTO

1.2 Descrizione del problema

Questa Tesi si propone di affrontare un problema di ottimizzazione legato alla gestione dei dati nella piattaforma Internet of Thing (IoT) IOtto.

1.3 Obiettivi di Tesi

L'obiettivo della Tesi risulta essere quello di realizzare un prototipo di File System (FS) che sia scalabile, in grado di effettuare rapide letture e scritture, strutturalmente solido e ottimizzato nell'allocazione di spazio.

In particolare, il lavoro si concentra sulla progettazione e realizzazione di un prototipo alternativo al sistema esistente, con l'intento non solo di mantenere le prestazioni attuali, ma, ove possibile, di migliorarle.

1.3.1 Domande

- Quali tipi di dati devono essere memorizzati e gestiti?
- Qual è la quantità stimata di dati da gestire a regime?
- Quali saranno le dimensioni massime dei file e la granularità delle operazioni sui dati?
- Come garantire un basso consumo di risorse?
- Qual è la struttura di archiviazione più adatta?
- Il file system dovrà scalare per gestire un numero crescente di dispositivi IoT?
- Qual è la strategia per l'espansione dello spazio di archiviazione?
- Come gestire guasti hardware o interruzioni improvvise di alimentazione?
- È necessario prevedere meccanismi di backup o snapshot per i dati?
- Come si possono introdurre nuove funzionalità senza compromettere i dati esistenti?

1.3.2 Scopo

L'obiettivo finale è quello di apportare un significativo miglioramento al software esistente, sfruttando l'hardware già disponibile per ottenere prestazioni più elevate. Questo intervento mira a incrementare le capacità del prodotto, garantendo una maggiore efficienza e un'esperienza d'uso più soddisfacente per i clienti.

In particolare, il miglioramento si traduce nell'ottimizzazione delle funzionalità attuali e nell'ampliamento delle possibilità offerte dal sistema, senza la necessità di modificare o sostituire l'hardware esistente. Questo approccio consente di massimizzare il valore del prodotto, andando incontro alle crescenti aspettative degli utenti e mantenendo un vantaggio competitivo sul mercato.

In questo contesto, la struttura dati B-Tree rappresenta una soluzione promettente per migliorare le prestazioni dei FS, grazie alla sua capacità di organizzare e recuperare i dati in modo efficiente.

Tuttavia, l'applicazione del B-Tree ai FS per piattaforme IoT rimane un campo relativamente poco esplorato e complesso nella sua realizzazione. La presente Tesi si propone di colmare questa lacuna attraverso lo studio e la realizzazione di un prototipo di FS basato sul B-Tree.

L'obiettivo è dimostrare come questa struttura dati possa essere utilizzata per ottimizzare le operazioni di memorizzazione e recupero dei dati, migliorando al contempo l'efficienza complessiva del sistema.

1.4 Struttura della Tesi

Questa Tesi è strutturata in tre capitoli principali.

Il primo capitolo fornisce le basi teoriche del lavoro, offrendo una panoramica sui moderni FS e sulla loro connessione con il mondo dell'IoT. Viene inoltre approfondita la struttura dati B-Tree, evidenziandone le proprietà principali e la sua rilevanza rispetto al problema affrontato.

Il secondo capitolo si concentra sulla progettazione e sull'implementazione del prototipo di FS. Vengono descritte le scelte effettuate durante il processo di sviluppo e spiegato come il B-Tree sia stato utilizzato per soddisfare le esigenze specifiche della piattaforma IOtto.

Infine, il terzo capitolo discute i risultati ottenuti dalla valutazione del prototipo. Vengono inoltre presentate le conclusioni tratte dallo studio e delineate le possibili direzioni per future modifiche e sviluppi in questo campo.

Chapter 2

State of the art

I suggest referencing stuff as follows: ?? or ??

2.1 Onit e IOtto

2.2 File System nell'IoT

In genere, i dispositivi IoT hanno capacità di archiviazione dei dati limitate. Per questo motivo, la maggior parte dei dati acquisiti viene trasmessa utilizzando protocolli di comunicazione come MQTT o CoAP, per essere poi elaborata e archiviata. Ma non è questo il problema principale.

Quando si tratta dei dati dell'IoT, si parla della necessità di gestire dati eterogenei, di trasformarli, aggregarli, analizzarli e integrarli per renderli pronti alle fasi successive di analisi, mantenendone nel contempo integrità e riservatezza e senza compromettere né le prestazioni di sistema, né l'affidabilità, né la scalabilità, né la flessibilità e, ancor più, senza aggravare significativamente i costi. [?]

2.2.1 Eterogeneità

L'ecosistema dell'IoT è caratterizzato dalla presenza di numerosi organismi di standardizzazione, ciascuno dei quali contribuisce con un ampio ventaglio di norme e protocolli che spesso si sovrappongono o risultano in conflitto tra loro.

Questa diversità si riflette nei dati acquisiti dai dispositivi IoT, che vengono prodotti in una combinazione di formati, inclusi dati strutturati, semi-strutturati e non strutturati, generando ulteriori complessità.

L'eterogeneità nei formati e nei protocolli non si limitano alla fase di acquisizione, ma hanno un impatto diretto anche nelle fasi di archiviazione, complicando la gestione dei dati nei sistemi di storage.

Così come non possiamo pensare di risolvere la gestione dei dati IoT come avremmo fatto con un data warehouse negli anni '90, dobbiamo cercare nuove soluzioni per gestire i differenti tipi di dati e la mole di quest'ultimi.

In molti casi, è opportuno archiviare non il dato grezzo, bensì il dato già elaborato, soprattutto in tutti quegli scenari nei quali i dati stessi devono essere analizzati più di una volta.

2.3 B-Tree

Citando

B-trees are balanced search trees designed to work well on disk drives or other direct-access secondary storage devices. B-trees are similar to red-black trees, but they are better at minimizing the number of operations that access disks.

2.4 B-Tree nella realizzazione dei File System

2.5 Some cool topic

Chapter 3

Contribution

You may also put some code snippet (which is NOT float by default), eg: chapter 3.

3.1 Fancy formulas here

```
1 public class HelloWorld {
2     public static void main(String[] args) {
3         // Prints "Hello, World" to the terminal window.
4         System.out.println("Hello, World");
5     }
6 }
```

Bibliography

[CLRS22] Thomas H Cormen, Charles E Leiserson, Ronald L Rivest, and Clifford Stein. *Introduction to algorithms*. MIT press, 2022.

Acknowledgements

Optional. Max 1 page.