

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

MASTER THESIS

---

# Relaxed Radix Balanced Trees as Immutable Vectors Scala

---

*Author:*

Nicolas STUCKI

*Supervisor:*

Vlad URECHE

*A thesis submitted in fulfilment of the requirements  
for the degree of Master in Computer Science*

*in the*

LAMP  
Computer Science

December 2014

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

# *Abstract*

School of Computer and Communications  
Computer Science

Master in Computer Science

## **Relaxed Radix Balanced Trees as Imutable Vectors Scala**

by Nicolas STUCKI

The Thesis Abstract is written here (and usually kept to just this page). The page is kept centered vertically so can expand into the blank space above the title too...

# Contents

<b>Abstract</b>	<b>i</b>
<b>Contents</b>	<b>ii</b>
<b>List of Figures</b>	<b>iv</b>
<b>List of Tables</b>	<b>v</b>
<b>Abbreviations</b>	<b>vi</b>
<b>1 Introduction</b>	<b>2</b>
1.1 Main Section 1 . . . . .	2
1.1.1 Subsection 1 . . . . .	2
1.1.2 Subsection 2 . . . . .	2
1.2 Main Section 2 . . . . .	2
<b>2 Vector Structure</b>	<b>3</b>
2.1 Radix Balanced Vectors . . . . .	3
2.1.1 Tree structure . . . . .	3
2.1.2 Operations . . . . .	3
2.1.2.1 Apply (get element at index) . . . . .	3
2.1.2.2 Updated . . . . .	3
2.1.2.3 Additions . . . . .	3
Append and Prepend . . . . .	3
Concatenation and Insert . . . . .	3
2.1.2.4 Splits . . . . .	4
2.2 Parallel Vectors . . . . .	4
2.2.1 Splitter Iterator . . . . .	4
2.2.2 Combiner Builder . . . . .	4
2.3 Relaxed Radix Balanced Vectors . . . . .	4
2.3.1 Tree structure . . . . .	4
2.3.2 Operations . . . . .	4
2.3.2.1 Apply (get element at index) . . . . .	4
2.3.2.2 Updated . . . . .	4
2.3.2.3 Additions . . . . .	4
Append and Prepend . . . . .	4
Insert . . . . .	4

---

Concatenation . . . . .	4
2.3.2.4 Splits . . . . .	4
<b>3 Implementation and Optimizations</b>	<b>5</b>
3.1 Where does time go? . . . . .	6
3.1.1 Arrays . . . . .	6
3.1.2 Computing indices . . . . .	6
3.2 Displays . . . . .	6
3.2.1 As cache . . . . .	6
3.2.2 For transient states . . . . .	6
3.3 Builder . . . . .	6
3.4 Iterator . . . . .	6
3.5 Relaxing the Radix . . . . .	6
3.5.1 Displays . . . . .	6
3.5.2 Builder . . . . .	6
3.5.3 Iterator . . . . .	6
<b>4 Performance</b>	<b>7</b>
4.1 Main Section 1 . . . . .	7
4.1.1 Subsection 1 . . . . .	7
4.1.2 Subsection 2 . . . . .	7
4.2 Main Section 2 . . . . .	7
<b>5 Testing</b>	<b>8</b>
5.1 Teststing correctness . . . . .	8
5.1.1 Invariant Assertions . . . . .	8
5.1.2 Unit tests . . . . .	8
5.2 Main Section 2 . . . . .	8
<b>6 Related Work</b>	<b>9</b>
6.1 RRB-Vectors in Clojure . . . . .	9
<b>7 Conclusions</b>	<b>10</b>

# List of Figures

# List of Tables

# Abbreviations

<b>JIT</b>	<b>J</b> ust <b>I</b> n <b>T</b> ime
<b>RB</b>	<b>R</b> adix <b>B</b> alanced
<b>RRB</b>	<b>R</b> elaxed <b>R</b> adix <b>B</b> alanced

I



# Chapter 1

## Introduction

### 1.1 Main Section 1

#### 1.1.1 Subsection 1

#### 1.1.2 Subsection 2

### 1.2 Main Section 2

I

## Chapter 2

# Vector Structure

### 2.1 Radix Balanced Vectors

#### 2.1.1 Tree structure

#### 2.1.2 Operations

##### 2.1.2.1 Apply (get element at index)

##### 2.1.2.2 Updated

##### 2.1.2.3 Additions

#### Append and Prepend

#### Concatenation and Insert

#### 2.1.2.4 Splits

## 2.2 Parallel Vectors

### 2.2.1 Splitter Iterator

### 2.2.2 Combiner Builder

## 2.3 Relaxed Radix Balanced Vectors

### 2.3.1 Tree structure

### 2.3.2 Operations

#### 2.3.2.1 Apply (get element at index)

#### 2.3.2.2 Updated

#### 2.3.2.3 Additions

### Append and Prepend

### Insert

### Concatenation

#### 2.3.2.4 Splits

### I



## Chapter 3

# Implementation and Optimizations

### 3.1 Where does time go?

#### 3.1.1 Arrays

#### 3.1.2 Computing indices

### 3.2 Displays

#### 3.2.1 As cache

#### 3.2.2 For transient states

### 3.3 Builder

### 3.4 Iterator

### 3.5 Relaxing the Radix

#### 3.5.1 Displays

#### 3.5.2 Builder

#### 3.5.3 Iterator



## Chapter 4

# Performance

### 4.1 In practice: Running on JVM

#### 4.1.1 Cost of Abstraction and JIT Inline

### 4.2 Measuring performance

### 4.3 Generators

### 4.4 Benchmarks

#### 4.4.1 Apply

#### 4.4.2 Concatenation

#### 4.4.3 Append

#### 4.4.4 Prepend

#### 4.4.5 Iterator

#### 4.4.6 Builder

#### 4.4.7 Parallel split-combine

## Chapter 5

# Testing

### 5.1 Teststing correctness

#### 5.1.1 Invariant Assertions

#### 5.1.2 Unit tests

### 5.2 Main Section 2

I



## Chapter 6

# Related Work

### 6.1 RRB-Vectors in Clojure

I

## Chapter 7

## Conclusions