Imperfectly Nested Loops (Section 6.4 and 6.5)

Anshuman Dhuliya
144050002

November 4, 2015

Contents

1 Introduction

This report is an attempt to explain optimizations in presence of imperfectly nested loops in a crisp and clear way. It has been adapted from the sections 6.4 and 6.5 of the book [1].

2 A Short Overview

When an imperfectly loop nest (Section 3) is encountered, the actions taken can be categorised into the following cases:

An algorithmic approach is discussed ahead

Case 1 *When the outermost loop can be parallelized*, parallelize the loop and move further into its body to look for more optimizations.

Case 2 If the *outermost loop cannot be parallelized*, then maximally distributing it around the statements in its body, can be an effective transformation. This step will create further loop nests, which may lead to the following cases:

Case 2.1 *Perfectly nested loop nests.* In this case use the perfect nest loop optimization algorithm.

Case 2.1 *Imperfectly nested loop nests.* This is a result of a *tight recurrence* (a cyclic dependency) involving a statement and an inner loop. In this case it is best to leave the loop sequential and move into its body to look for other optimizations.

Case 3 *The outer loop can neither be parallelized nor distributed.* Then leave the loop sequential and move into its body to look for other possible optimizations.

3 What is an Imperfect Loop Nest

4 What is an Imperfect Loop Nest

References

[1] R. Allen and K. Kennedy, *Optimizing Compilers for Modern Architectures.* Maurgan Kaufmann Publishers, 1985.