



Mahidol University
International College

ICCS315 Applied Algorithms

False Sharing

Written by
Archer N. Phillips, Krittin Nisunarat 6280782

9 April 2023

Abstract

We can not deny that cache protocols are essential for improving the performance of computer systems in several ways. One essential way is by reducing the time it takes to access data from memory. However, the restriction is the size of it with a trade off. Obviously, any kinds of volatile memories have a small size, but fast storage area that stores frequently used data and instructions so that they can be accessed quickly by the processor. However, the performance is uncertain as some data can be stored on the same address on share stage cache line. This situation is well-known in computer science that is called **false sharing**.

False sharing is essentially a performance-degrading usage pattern that can arise in systems with distributed, coherent caches at the size of the smallest resource block managed by the caching mechanism. When a system participant attempts to periodically access data that is not being altered by another party, but that data shares a cache block with data that is being altered, the caching protocol may force the first participant to reload the whole cache block despite a lack of logical necessity. The caching system is unaware of activity within this block and forces the first participant to bear the caching system overhead required by true shared access of a resource. This can happen several ways which will be discussed in this paper after the experiment.

In this paper, MESIF cache coherence protocol will be discussed in detail and stage management of the cache protocol in multiprocessor. In addition, existing solution will be illustrated as many of architectures and programming languages are integrated. Finally, our false sharing solution will be determined at the end of the paper.

Contents

1	Cache Protocol	2
2	Existing Solution	3
3	Theoretical Solution	4

Chapter 1

Cache Protocol

Chapter 2

Existing Solution

Chapter 3

Theoretical Solution

Reference