

Жибарь Марк

Лабораторная работа №2 «Оптимизация работы параллельной  
программы с кэш памятью»

## Оглавление

1. Цель работы .....	3
2. True и false sharing .....	3
3. Ход работы .....	5
4. Выводы.....	9
Литература.....	9

## **1. Цель работы**

Получить навык оптимизации данных между потоками в работе с кэш памятью, изучение двух типов разделения кэш линий: true и false sharing.

## **2. True и false sharing**

Обращение (с модификацией) потоков к данным структур, расположенных очень близко друг к другу, а именно в блоке, равном длине одной кэш-линии (64 байт) называется Cache Line sharing. Существует два типа разделения кэш-линий: True sharing и False sharing.

True sharing (истинное разделение) – это когда потоки имеют доступ к одному и тому же объекту памяти, например, общей переменной или примитиву синхронизации.

False sharing происходит, когда потоки на разных процессорах изменяют переменные, которые находятся в одной и той же строке кэша. Это делает недействительной строку кэша и вынуждает обновление памяти поддерживать когерентность кэша.

На рисунке 1 потокам 0 и 1 требуются переменные, которые находятся рядом в памяти и находятся в одной строке кэша. Строка кэша загружается в кэши CPU 0 и CPU 1 (серые стрелки). Несмотря на то, что потоки изменяют различные переменные (красные и синие стрелки), строка кэша становится недействительной, что вынуждает обновление памяти поддерживать когерентность кэша.

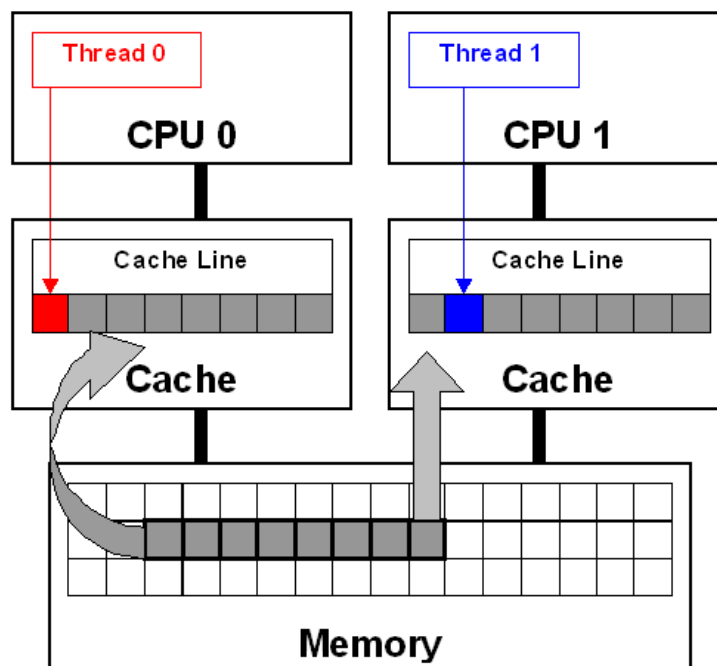


Рисунок 1 – Доступность общей памяти из разных потоков

Чтобы обеспечить согласованность данных в нескольких кэшах, процессоры Intel® с поддержкой нескольких процессоров следуют протоколу MESI (Modified / Exclusive / Shared / Invalid). При первой загрузке строки кэша процессор помечает строку кэша как «Эксклюзивный» доступ. Пока строка кэша помечена как исключительная, последующие загрузки могут свободно использовать существующие данные в кэше. Если процессор видит ту же строку кэша, загруженную другим процессором на шине, он помечает строку кэша с доступом «Shared». Если процессор хранит строку кэша, помеченную как «S», строка кэша помечается как «Модифицированная», и всем другим процессорам отправляется сообщение «Недопустимая» строка кэша. Если процессор видит ту же строку кэша, которая теперь помечена как «М», доступ к которой осуществляется другим процессором, процессор сохраняет строку кэша обратно в память и помечает свою строку кэша как «Общая». Другой процессор, который получает доступ к той же строке кэша, получает ошибку кэша.

### 3. Ход работы

В эксперименте будет производиться перемножение содержимых ячеек двух массивов и определяться время выполнения программы. После каждого запуска будет изменяться размерность массивов, чтобы менять трудоёмкость задачи. Также будут изменяться режимы распределения порций данных по потокам. В одном случае без разбиения итераций, в другом с разбиением их по блокам, количество которых будет равняться числу потоков, что будет соответствовать False и True sharing.

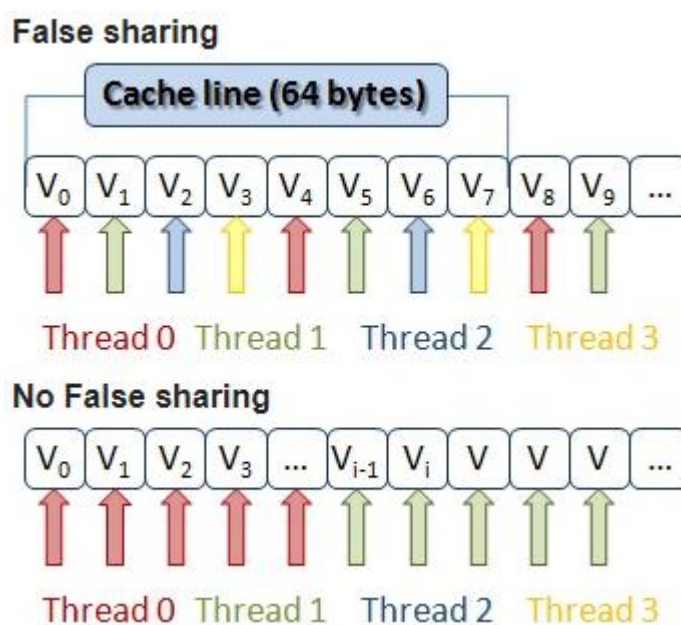


Рисунок 2 – Механизм False sharing и True sharing

В данной задаче, общими для всех потоков (shared) будут являться массивы A и B и переменная со значением их размерности.

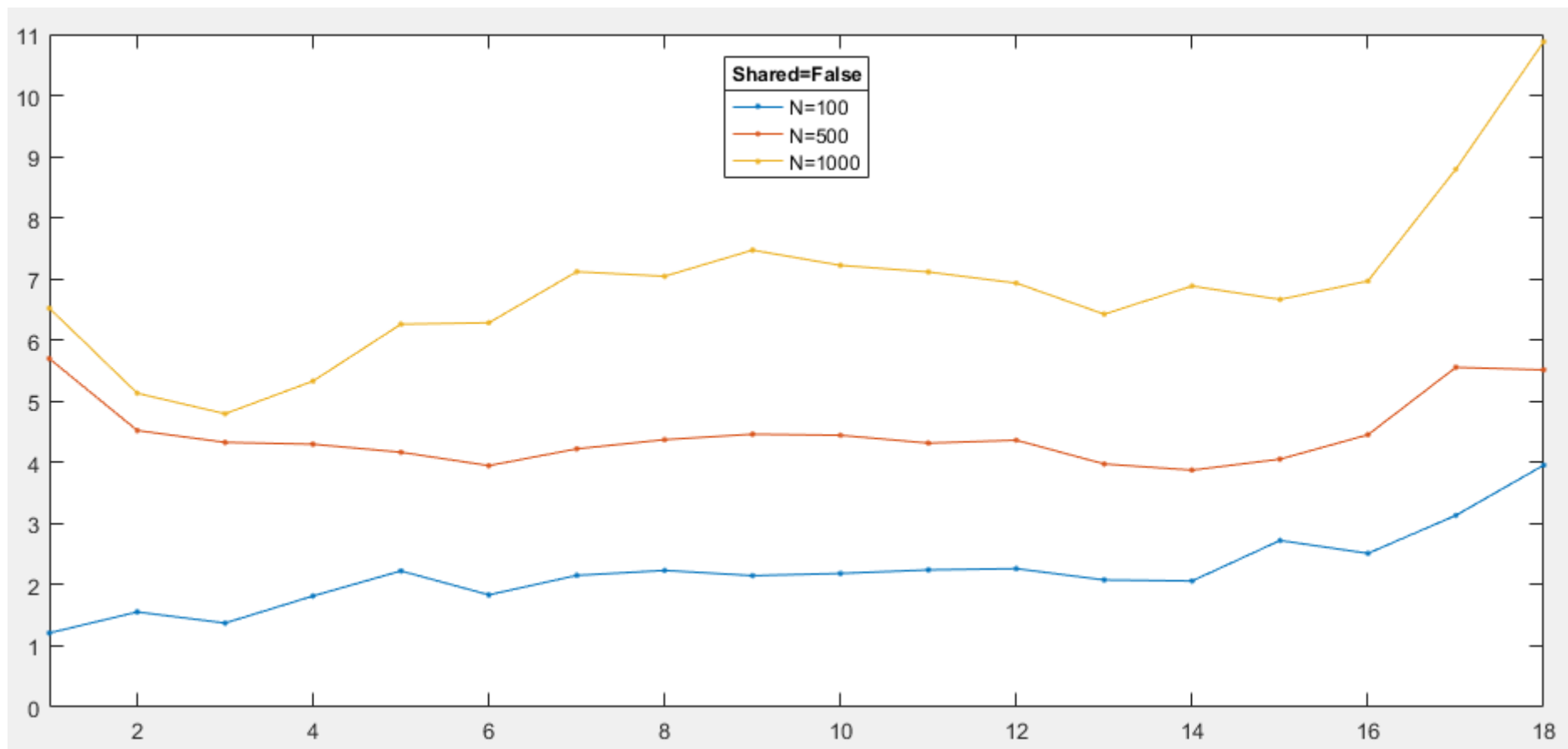


Рисунок 3 – Время выполнения программы без разбиения итераций по блокам (False sharing)

С False sharing увеличение масштабируемости программы только повышает её выполнение, что подтверждает проблему доступа потоков к общей области памяти.

Исправим этот недостаток разбиением итераций на блоки.

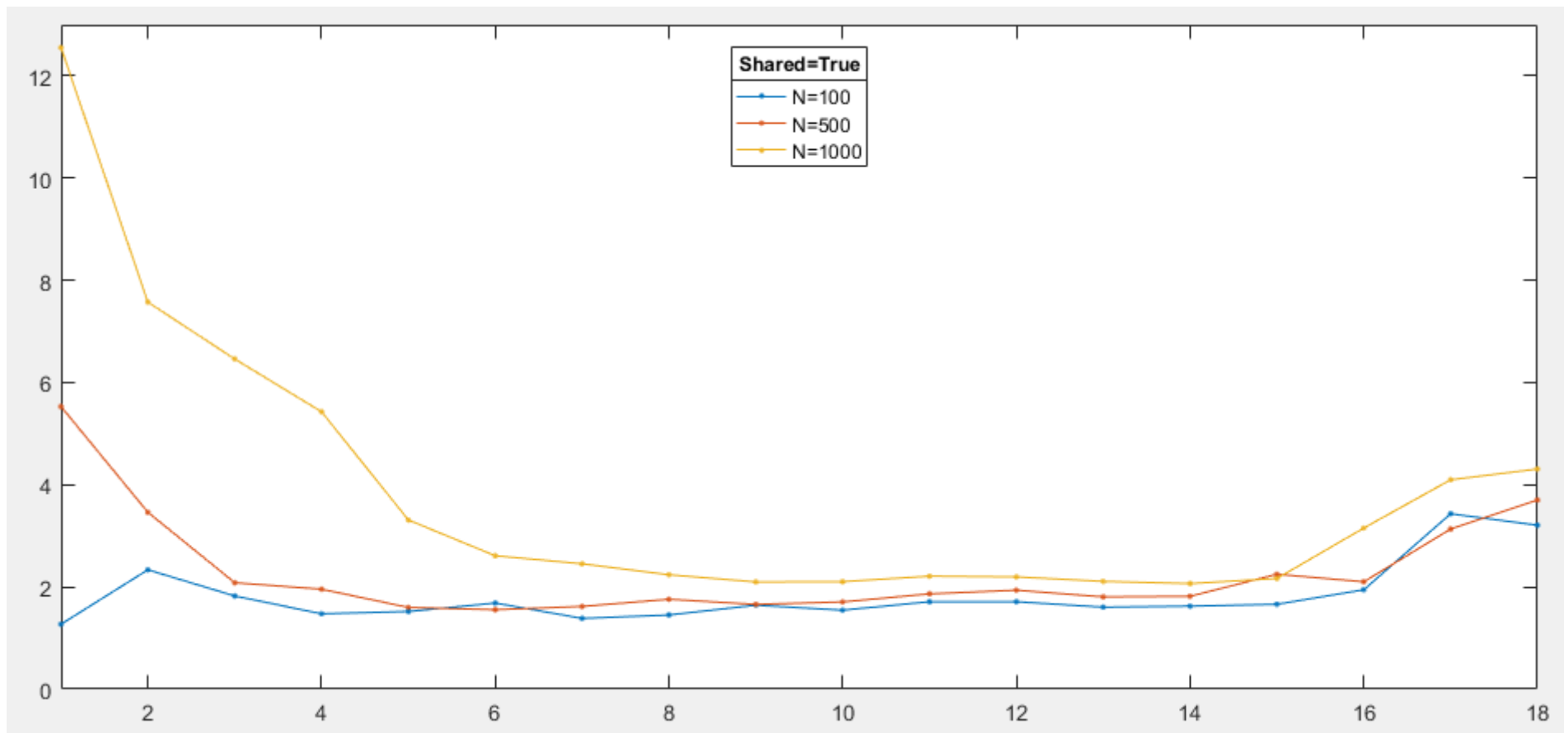


Рисунок 4 – Время выполнения программы с разбиением итераций по блокам (True sharing)



При True sharing программа стала хорошо масштабируемой, что особенно заметно на кривой большой трудоёмкости (размерность массивов  $N=1000$ ). Но при этом, в вычислениях с наименьшим числом итераций при размерности  $N=100$ ), True sharing, повышение масштабирования существенных изменений не даёт.

#### **4. Выводы**

В работе было изучено влияние механизмов доступа к памяти на время выполнения программы в зависимости от режимов разделения кэш линий. Было подтверждено влияние True sharing на масштабируемость программы при высокой трудоёмкости задачи.

#### **Литература**

- [1] Делиться не всегда полезно: оптимизируем работу с кэш-памятью:  
<https://habr.com/company/intel/blog/143446/>
- [2] Avoiding and Identifying False Sharing Among Threads:  
<https://software.intel.com/en-us/articles/avoiding-and-identifying-false-sharing-among-threads>