Софийски Университет

„Св. Климент Охридски”

Системи за паралелна обработка 2010

Проект на тема „Построяване на честотна таблица на входен поток от информация чрез използване на многонишкова програма”

Иван Стоилов

Компютърни Науки

Трети курс

Описание на задачата

Задачата, която трябваше да се реши е следната. Трябваше да се изготви честотна таблица на даден файл. Честотна таблица на файл представлява таблица, в която е записан броя на срещанията на всеки от символите, присъстващи във файла. Таблицата е нужна, за да може да се приложи алгоритъма на Хъфман за компресиране на файл. За оптимизиране на производителността на приложението работата трябваше да се разпредели в няколко нишки. Техният максимален брой се задава от потребителя при стартирането на програмата.

Реализация на проекта

За реализацията на проекта беше използван езика за програмиране Java. За разпределянето на работата на програмата в отделни нишки са използвани класовете *ThreadGroup*, Thread и *Runnable*.

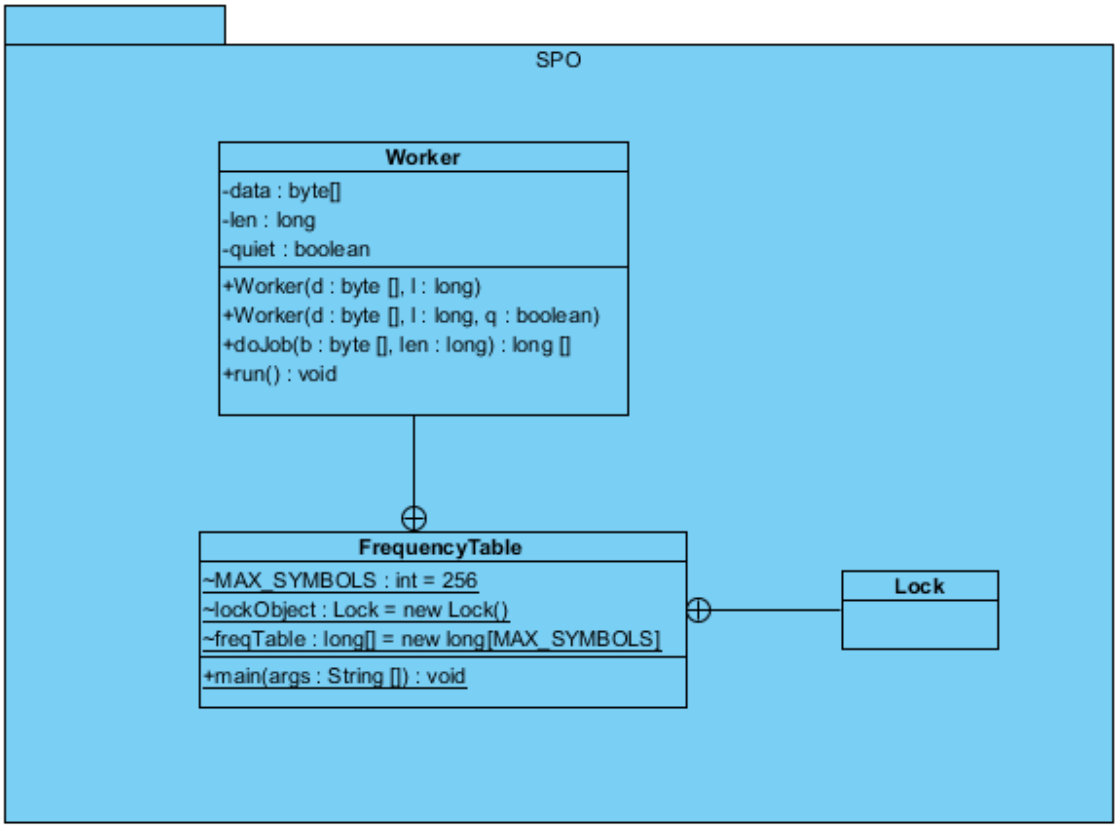
Описание на работата на приложението

След началното пускане на приложението се отваря файлът, който ще бъде обработван. След това започват да се създават и пускат нишки, който да обработват парчета от файла. Парчетата са с големина 1 MB. Нишките изготвят таблиците за частта от файла, която им е зададена. След това те обединяват получената таблица с главната таблица на файла. Това се повтаря докато не се достигне края на файла. Всички нишки се пускат в една и съща група (ThreadGroup) с име *huffman*. Максималния брой нишки е зададен от потребителя предварително. Програмата пуска толкова нишки колкото е максималния брой и чака някоя от тях да завърши работа. След като някоя нишка завърши работата си на нейно място се пуска нова. Така броя на активните нишки в групата се

поддържа постоянно. След като се достигне края на файла се изчакват всички нишки от групата да завършат.

Структура на приложението

Приложението се състой от 3 класа.



Класът FrequencyTable

Това е главният клас. Той отговаря за следните неща. Първо, той създава групата, в която се пускат нишките.

```
ThreadGroup threadPool = new ThreadGroup("huffman");
```

Второ, той отговаря за цикъла, който чете от файла и създава нишките.

```
while ((len = in.read(buf, 0, chunkSize)) != -1) {
   Thread t = new Thread(threadPool, new Worker(buf, len, quiet));
   t.start();

   while (threadPool.activeCount() == numThreads) {
        Thread.sleep(1);
   }
}
```

И трето, извежда начални съобщения и съобщение за времето, което е работила програмта.

Класът Worker

Този клас извършва основната част работата. Наследник е на класа *Runnable* и се състой от 2 метода. Методът *doJob* приема като параметри парче от паметта и създава честотна таблица по него.

```
public long[]  doJob(byte[] b, long len)
```

Методът *run* вика *doJob* и след това обединяват получената таблица с главната таблица на файла.

```
for (int i = 0; i < len; i++) {
   int key = (int)b[i] + 128;
   hash[key]++;
}
```

Методът *run* автоматично се извиква от нишката след като тя бъде стартирана.

```
public void run()
```

Класът Lock

Класът *Lock* служи за защита достъпа до общата памет – честотната таблица на файла.

Синхронизация на нишките

Общия за всички нишки ресурс е честотната таблица на файла. Тъй като може да се получи достъп до една и съща памет от различни нишки в едно и също време е необходимо те да бъдат синхронизирани. За синхронизация на нишките се използва класа *Lock.* Когато дадена нишка започне да обединява получените данни с общата таблица на файла тя заключва класа *Lock.* Ако във същото време друга нишка опита да достъпи общата таблица тя ще бъде блокирана докато първата не свърши работата си. По този начин нишките се изчакват и не може да се получат конфликти между тях.

```
synchronized (lockObject) {
   if (!quiet) {
        System.out.printf("Thread finished. Time %d ms.\n",
                          elapsedTimeMillis);
   }

   for (int i = 0; i < hash.length; i++) {
        freqTable[i] += hash[i];
   }
}
```

Индикация по време на работа

Режим на подробна информация

В този режим програмата извежда информационни съобщения по време на работа. В началото, преди да започне процеса на изготвяне на таблицата, се показват името на файла, големината му в байтове, максималния броят нишки, които ще бъдат стартирани и размера на парчетата от файла, които се подават на нишките. В процеса на работа всяка нишка извежда информационно съобщение

когато завърши работа с информация за това колко време е работила. След завършването на последната нишка се извежда съобщение, показващо времето за изпълнение на цялата програма.

Режим на съкратена информация

В този режим се извежда едно единствено съобщение, което показва времето за изпълнение на цялата програма. За да се премине в този режим, програмата трябва да бъде извикана с трети параметър “q”.

Потребителски интерфейс

За да се стартира програмата трябва да се извика с 2 или 3 входни параметъра.

Синтаксис: java –jar FrequencyTable.jar <filename> <number of threads> [q].

<filename> - задава пътят до файла, който чиято честотна таблица ще се търси (пълен или релативен)

<number of threads> - задава максималния брой нишки, който да се използват

<quiet> - пуска програмата в quiet режим (без да се извежда подробна информация)

Сравнителни данни от тестове

Тест 1

Характеристики на процесора на машината

Модел на процесора: **Intel® Core 2 Duo® T5450**

Кеш: **2 MB**

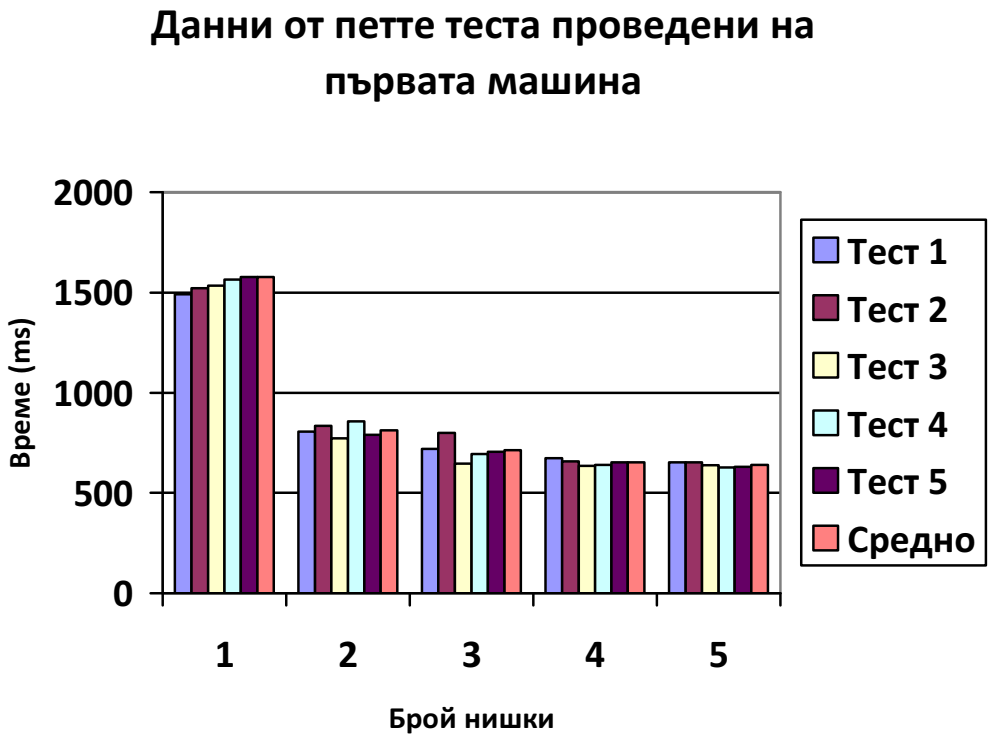
Работна честота: **1.66 GHz**

Честота на шината: **667 MHz**

Големина на файла

Използван беше файл с големина **93.5 MB**. Времената са в милисекунди.

Брой нишки	1	2	3	4	5
Тест 1	1491	807	720	673	652
Тест 2	1521	834	798	657	652
Тест 3	1534	772	646	636	638
Тест 4	1565	857	694	641	626
Тест 5	1577	789	706	652	630
Средно	1577	812	713	652	640



Тест 2

Характеристики на процесора на машината

Модел на процесора: **Intel® Xeon® E5520**

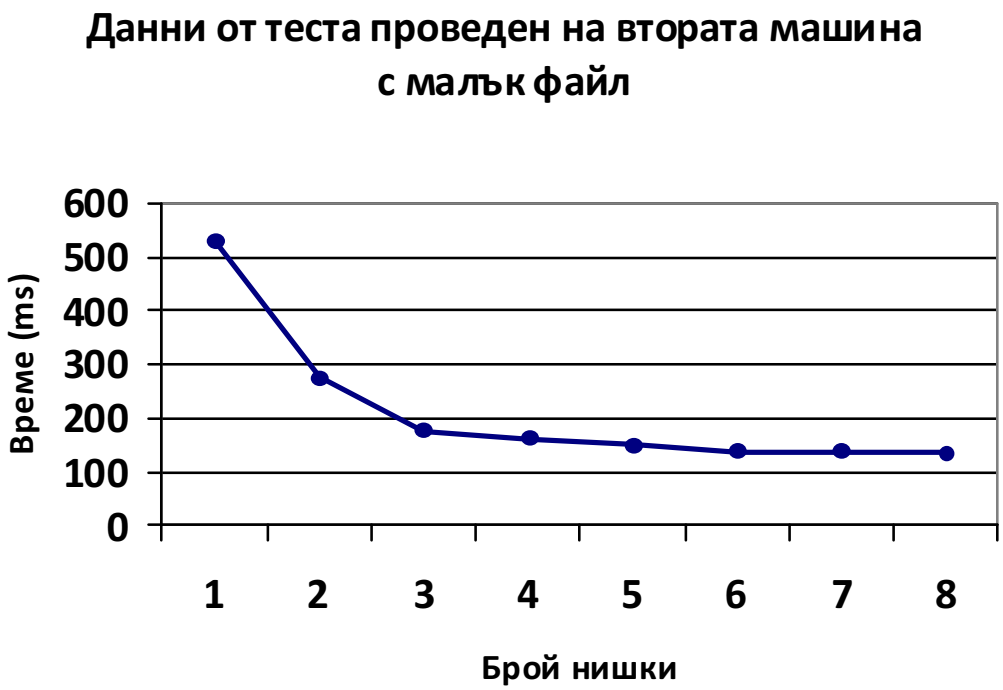
Кеш: **8 MB**

Работна честота: **2.27GHz**

Големина на файла

Използван беше файл с големина **131.44 MB**. Времената са в милисекунди.

Брой нишки	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Средно	529	275	176	162	149	138	137	136



Тест 3

Характеристики на процесора на машината

Модел на процесора: **Intel® Xeon® E5520**

Кеш: **8 MB**

Работна честота: **2.27GHz**

Големина на файла

Използван беше файл с големина **696.60 MB**. Времената са в милисекунди.

Брой нишки	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
Средно	2664	1484	1014	813	701	685	659	665

Данни от теста проведен на втората машина с голям файл