*Софийски Университет „Св. Климент Охридски“*

*Факултет по Математика и Информатика*

**Системи за паралелна обработка**

**Умножение на матрици**

**Проверил:**

Гл. Ас Христо Христов

**Изготвил:**

Мила Светлославова Русева,

ф.н. 80916, Компютърни науки, 3 курс, 1 група

**Условие:**

Разглеждаме матриците A с размерност (m, n) и B с размерност (n, k). Матрицата C=A.B, равна на произведението на A и B ще има размерност (m, k). Да се напише програма, която пресмята матрицата C. Работата на програмата по умножението на матриците да се раздели по подходящ начин на две или повече нишки (задачи). Изискванията към програмата са следните:

* Размерността на матриците се задава от подходящо избрани командни параметри – например „-m 1024 -n 512 -k 2048“;
* Елементите на матриците генерираме произволно с помощта на Math.random() или класа java.util.Random; (Тоест матриците може да имат float или double елементи)
* Друг команден параметър задава максималния брой нишки (задачи), на които разделяме работата по пресмятането елементите на C – например “–t 1” или “–tasks 3”;
* Извежда подходящи съобщения на различните етапи от работата си, както и времето отделено за изчисление;
* Да се осигури възможност за „quiet“ режим на работа на програмата, при който се извежда само времето отделено за изчисление на резултантната матрица, отново чрез подходящо избран друг команден параметър – например “-q”.

ЗАБЕЛЕЖКА:

* При желание за направата на подходящ графичен потребителски интерфейс (GUI) с помощта на класовете от пакета javax.swing задачата може да се изпълни от двама души; Разработването на графичен интерфейс не отменя изискването Вашата програма да поддържа изредените командни параметри. В този случай към функцията на параметъра параметъра “-q” се добавя изискването да не запуска графичния интерфейс. Причината за това е, Вашата програма да може да се тества отдалечено.
* Задачата може да се реши и с помощта на RMI (java.rmi). За целта трябва да се помисли за разпределения достъп до общия ресурс в случая матриците A, B и C.44

УТОЧНЕНИЯ:

* В условието на задачата се говори за разделянето на работата на две или повече нишки. Работата върху съответната задача на една нишка ще служи за еталон, по който да измерваме евентуално ускорение (T1). Тоест в кода реализиращ решенията на задачите трябва да се предвиди и тази възможност – задачата да бъде решавана от единствена нишка (процес); Пускайки програмата да работи върху задачата с помощта на единствена нишка, ще считаме че използваме серийното решение на задачата; Измервайки времето за работа на програмата при работа с „p“ нишки – Tp, изчисляваме Sp. Представените на защитата данни за работата на програмата, трябва да отразят и ефективността от работата и, тоест да се изчисли и покаже Ep.
* Командните аргументи (параметри) на терминална (конзолна) Java програма, получаваме във масива String args[] на main() метода, на стартовия клас. За „разбирането“ им (анализирането им) може да ползвате и външни библиотеки писани специално за тази цел . Един добър пример за това е: Apache Commons CLI (<http://commons.apache.org/cli/>).

**Тестване:**

Размерност на матриците, с които ще тестваме проекта:

*А512х4096 ; B4096x1024 ; C512x1024*

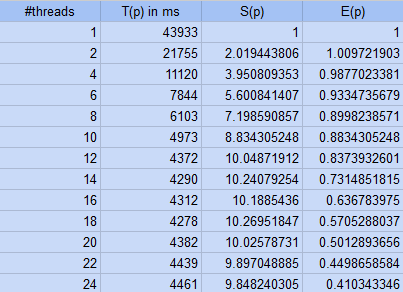
Като умножението им следва да бъде: AxB = C. Матриците са запълнени със случайни променливи от тип double с помощта на класа Math.random.

За смятането на изчислението и ускорението са използвани следните формулите:

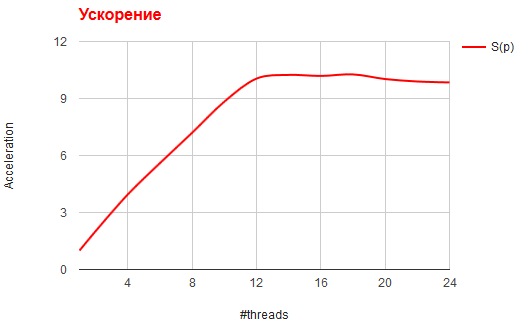
*S(p) = T(1)/T(p) – за ускорение*

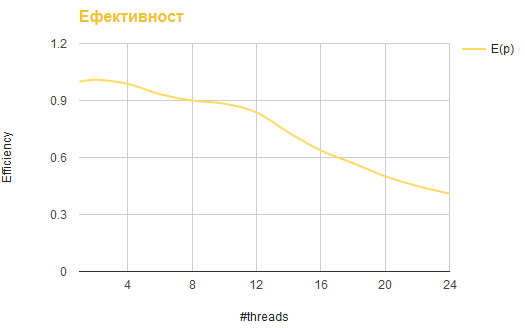
*E(p) = S(p)/p – за ефективност*

Без транспониране на матрицата B – четни нишки

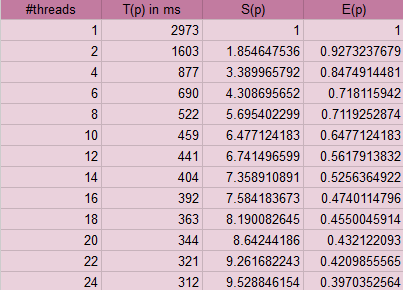




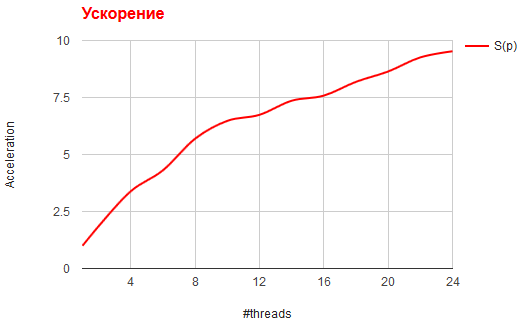


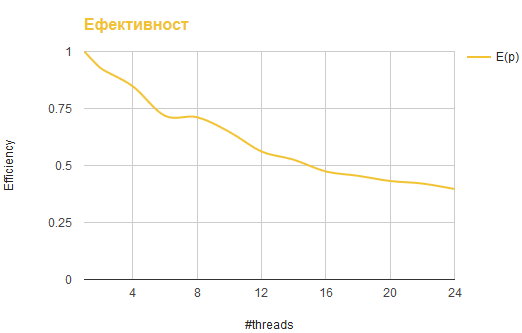


Транспонирана матрицата B – четни нишки

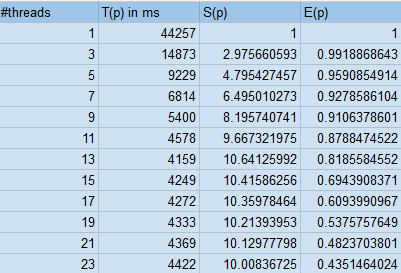




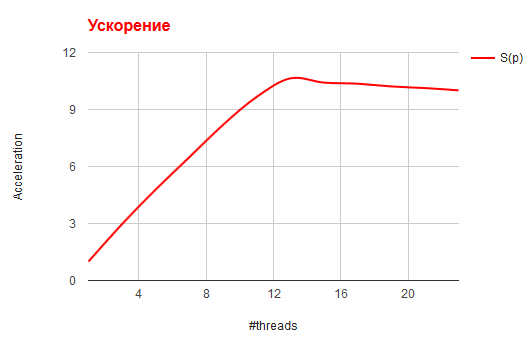


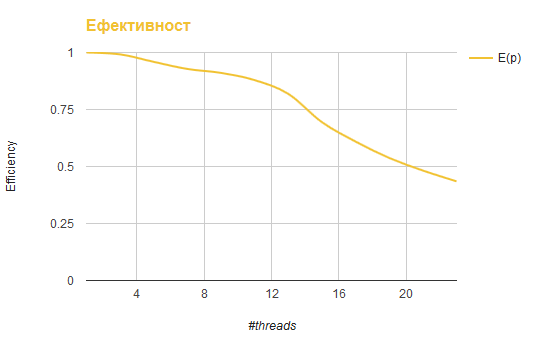


Без транспониране на матрица B – нечетни нишки









Транспонирана матрица B – нечетни нишки

