Технически университет – София. Факултет по компютърни системи и управление

Курсов работа по

Паралелно програмиране

**Паралелни процеси в Java.**

**Реализация в Cloud**

**Изготвил:**

Радослав Стефанов Досев ф.№ 121211127

Йордан Павлов Павлов ф.№ 121211031

26 януари 2013 г.

**Съдържание:**

1. Описание на проекта
2. Цел на проекта
3. Системни изисквания
4. Инсталация на софтуера
5. Изпълнение на операците
6. Реализация на проекта
7. Примерни резултати и скрийншоти
8. Анализ на тестовите резултати
9. **Описание на проекта**

Проекта представлява Java Enterprise приложение, инсталирано на апликейшън сървър (Tomcat 7), извършващо умножение над матрици. Умножението се извъшва по два начина, паралелно (чрез нишки) и праволинейно (чрез цикли). Матриците се подават, чрез “влачене” или избиране на текстови файлове съдържащи всеки по една матрица в себе си. Файловете са пращат към сървъра, парсват се и се прави умножението по двата различни подхода на изпълнение описани по горе. След изпълнението резултатът, както и времето нужно за различните подходи се връща под формата на текстови файл.

1. **Цел на проекта**

Целта на проекта е да се направи нагледна демонстрация на паралелизирани процеси, изпълнявани на отдалечен сървър, компилирани чрез езика Java.

Също така и да се направи статистика, сравнение и анализ на операциите изпълнявани паралелно и не паралелно.

1. **Системни изисквания**

- опреционна ситема Windows, Linux или Macintosh

- Java Runtime Environment 6

- Tomcat 7

- WEB браузър потдържащ HTML 5

**4. Инсталация на софтуера**

Инсталацията се идвършва в следния ред, като се предполага, че преди това е инсталирана опреционна ситема Windows, Linux или Macintosh.

1. Изтегля се и се инсталирва Java Runtime Environment 6 от следния адрес:

<http://www.oracle.com/technetwork/java/javase/downloads/jre6-downloads-1637595.html>

1. Изтегля и се инсталирва Tomcat 7 от следния адрес:

<http://tomcat.apache.org/download-70.cgi>

1. Изтегля и се инсталирва Tomcat.
2. Стартира се Tomcat чрес стартиране на **startup.sh** за Linux MAC или **startup.bat** за Windows, който се намира в \bin директорията.
3. Копира се matrix\_multiplication.war файла в \webapps директорията.

**5. Изпълнение на операциите**

**1.** Приложението се стартира, чрез стартиране на WEB браузър, потдържащ HTML 5 и въвеждане на адреса, сочещ към инсталирания апликейшън сървър.

Ако е инсталиран на същата машина това е <http://localhost:8080/PP_Project/>

**2.** Хващат се и влачат текстови файлове до полето “**Drop files** ot upload” или чрез клик на полето се избират ръчно

**3.**  Автоматично се изтегля текстови файл съдържащ резултата и информация за времето нужно за изчисление на резултата.

**6. Реализация**

Приложението се реализира с 3 Java класа.

*MatrixMultiplicator* – съдържа логиката за умножението на подадеди матрици, както с праволинеен така и с паралелен подход.

*ConcurrentMatrixMultiplyingTask* – съдържа логиката за всяка една нишка използвана при паралелното умножение на матрици. За всяка колона от парвата подадена матрица се прави отделна инстанция (нишка) на този клас.

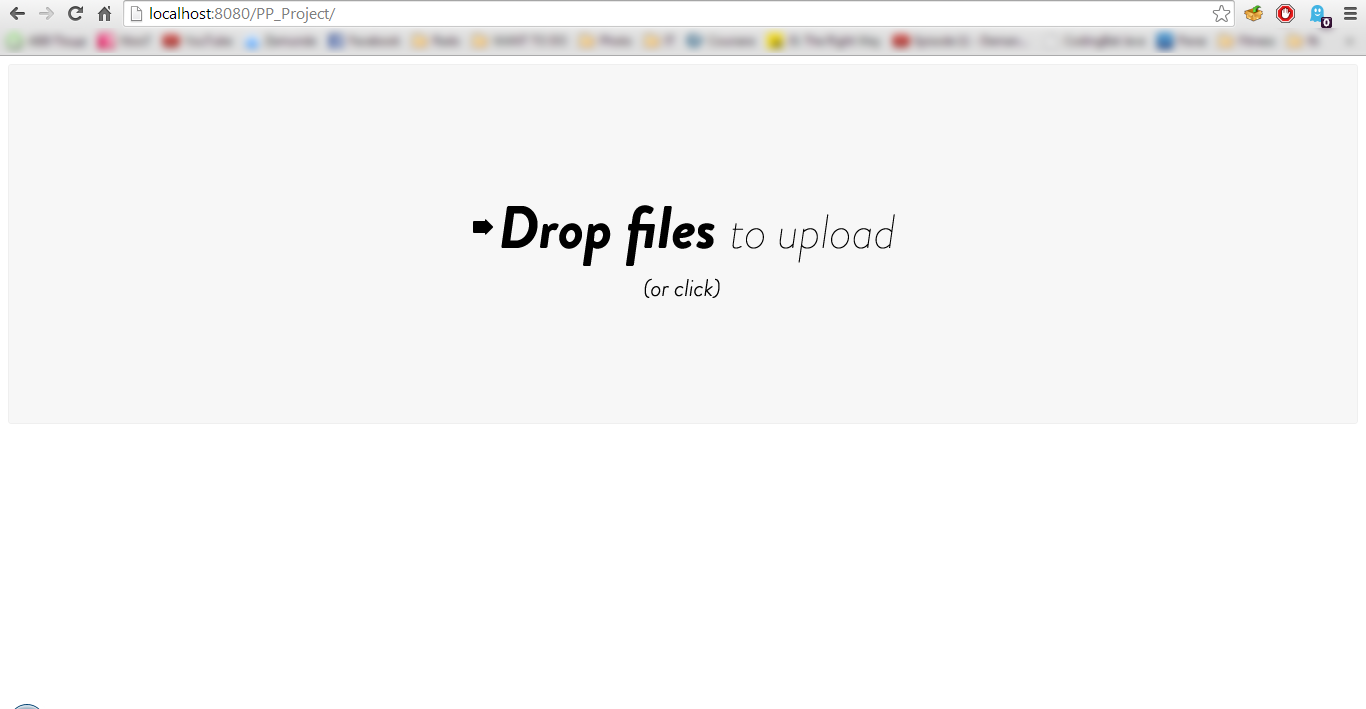
*FilesUploadServlet* – съдържа логиката свързваща входните заявки към сървара и навигираща процесите трицприложението

*MatrixFileGenerator* – съдържа логиката за генериране на тестови текстови файлове съдържащи матрици с произволни числа в тях

**7. Примерни резултати и скрийншоти.**

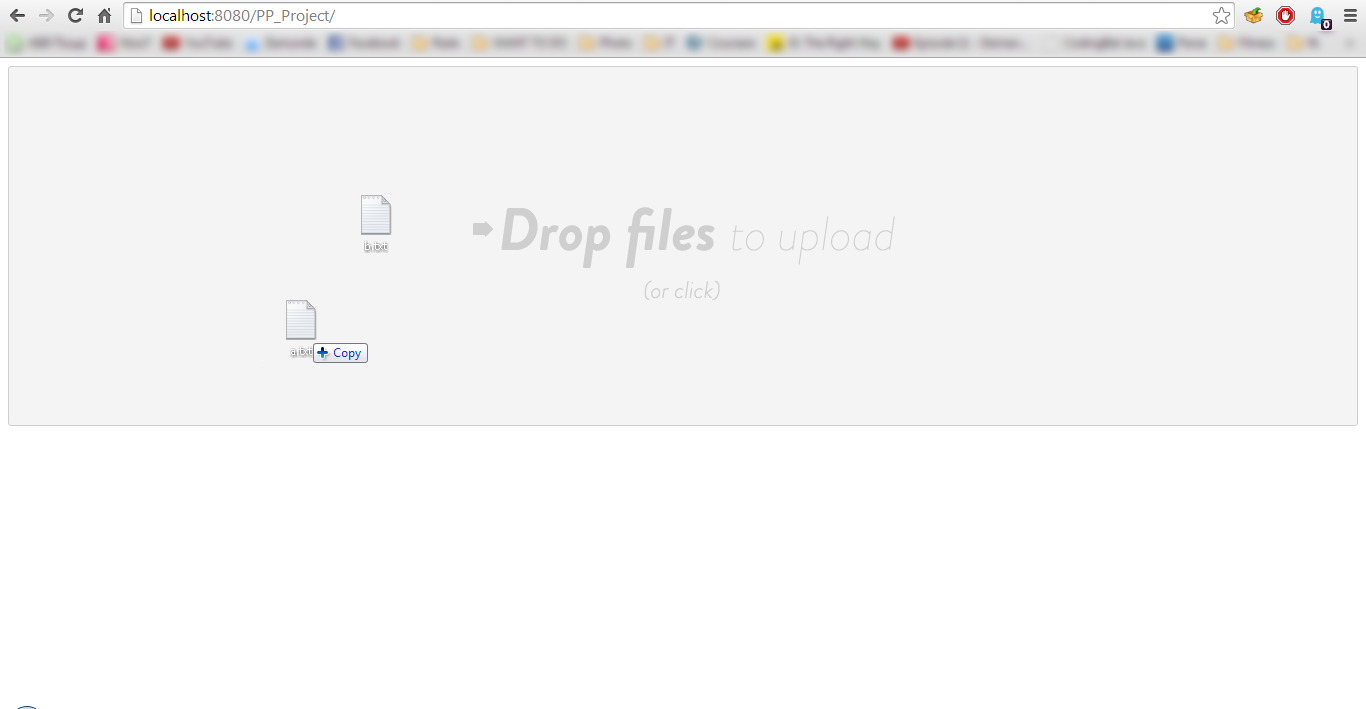
*начален екран*

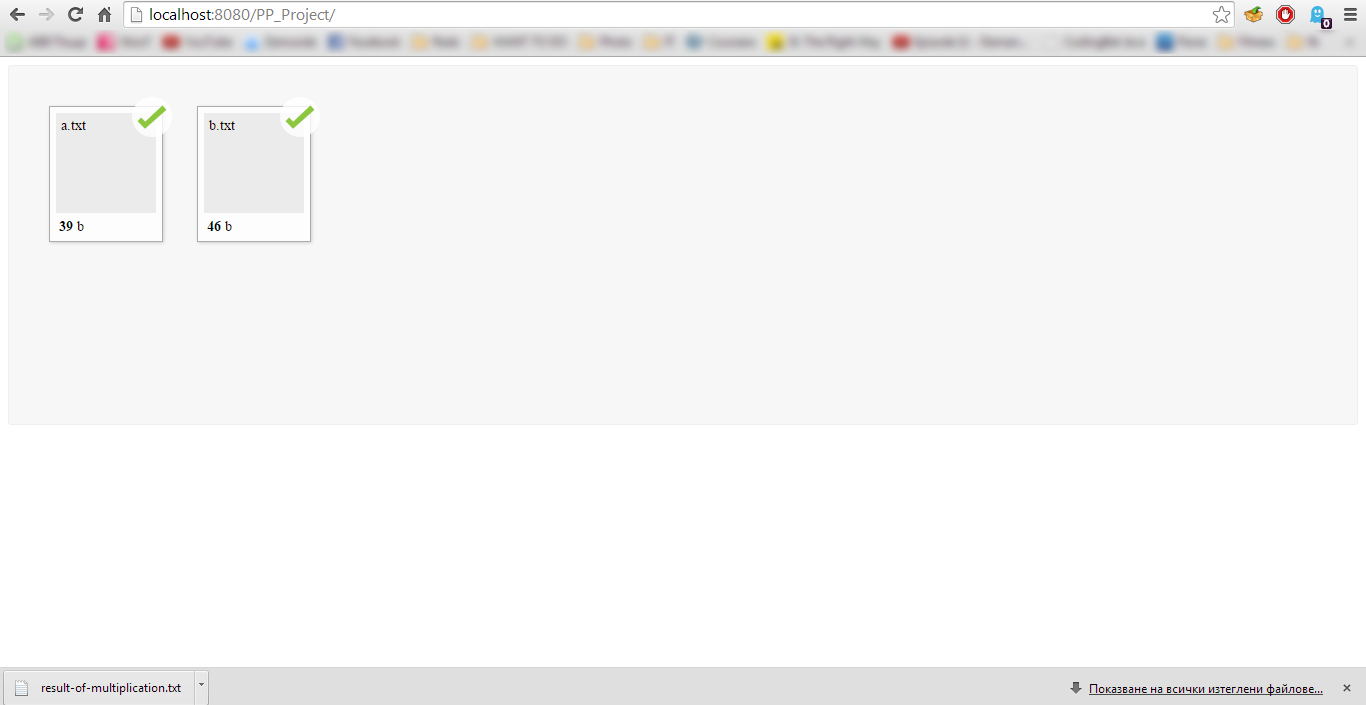
аа

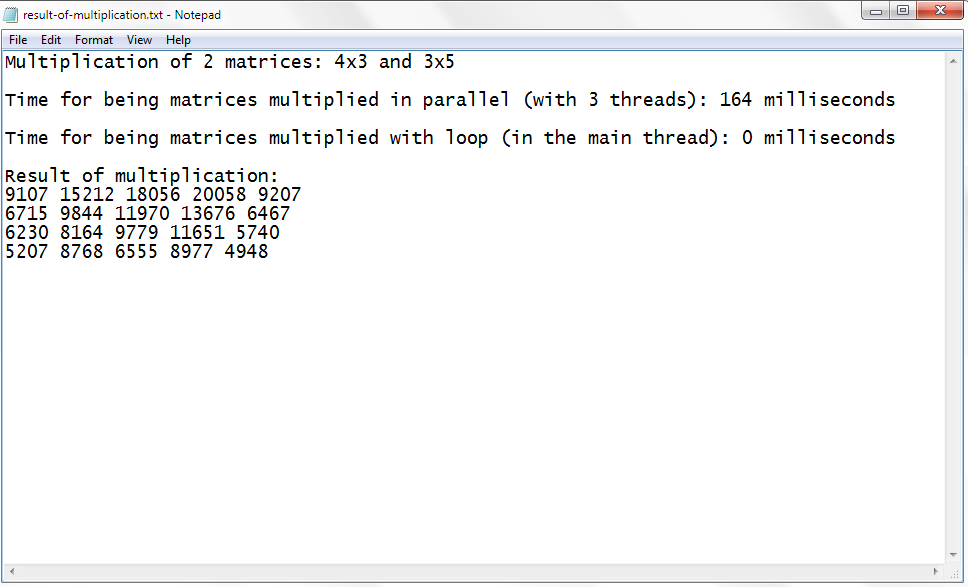


еишеиш

*влачене на файлове за изпращане към сървъра*



*файловете са успешно изпратени*

*примерни резултати*

8. Анализ на тестовите резултати