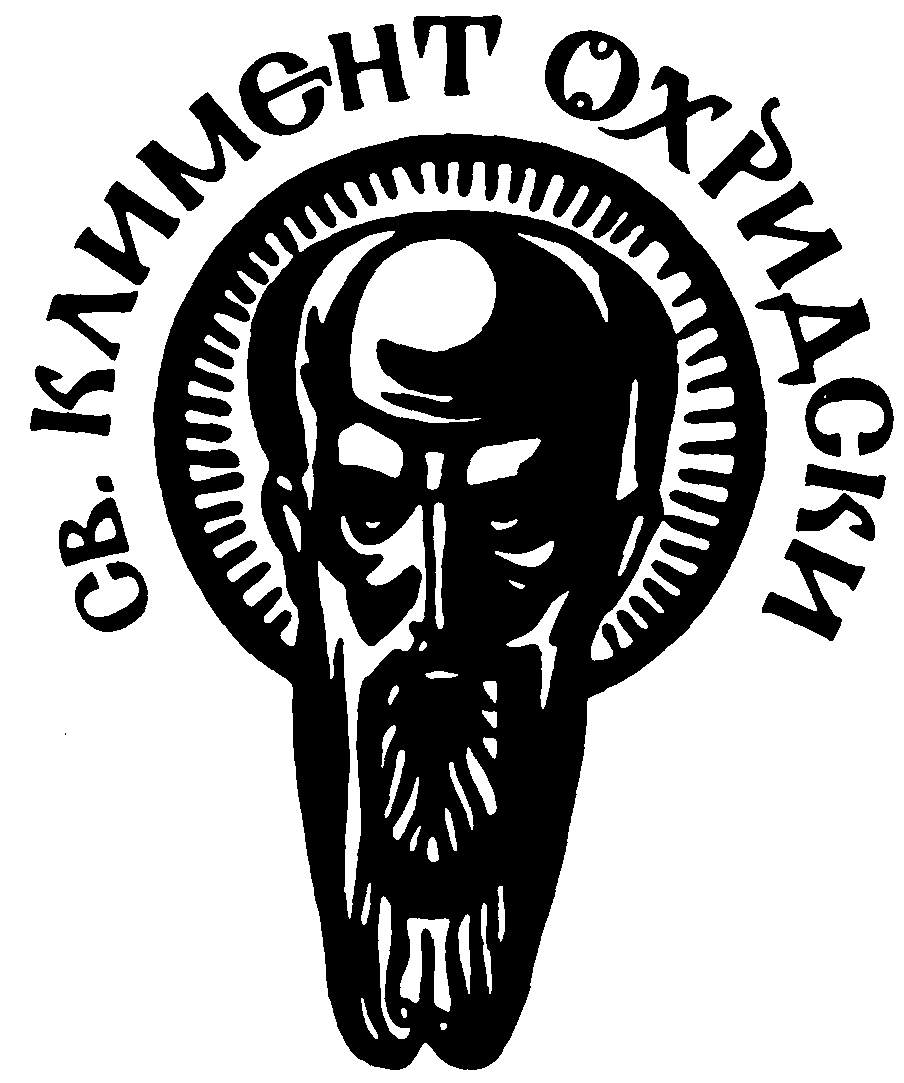
СОФИЙСКИ УНИВЕРСИТЕТ „СВ. КЛИМЕНТ ОХРИДСКИ“

ФАКУЛТЕТ ПО МАТЕМАТИКА И ИНФОРМАТИКА



ПРОЕКТ

ПО

„Мрежово програмиране“

НА ТЕМА 1, АЛГОРИТЪМ 1

Изготвил: Преподаватели:

Ричи Искренов Хаджиев доц. д-р Аделина Алексиева

№ 81450 доц. д-р Милен Петров

III курс, Компютърни науки

СЪДЪРЖАНИЕ

1. Тема
2. Описание на решението
3. Проектиране
4. Реализация
5. Ръководство за инсталация
6. Резултати
7. Използвана литература

1. ТЕМА

Клиент-сървър приложение чрез TCP протокола за търсене на зависимост на поведение на потребители.

Клиентът изпраща до сървър log файл със структура:

Time;Event context;Component;Event name;Description;Origin;IP address

Сървърът анализира поведението на потребителите на база данните от файла от колоната Event context с помощта на откриване на шаблон на данните чрез реализирането на **Mining Frequent Itemsets using the Apriori Algorithm** за извличането на данни.

Алгоритъма се опитва да покаже най-достъпваните ресурси от тип Event context в структурата на log файла.

2. ОПИСАНИЕ НА РЕШЕНИЕТО

Клиентът взима данните от входния файл и ги превръща в удобен за алгоритъма вид.

Изпраща ги на сървъра, който ги записва във файл. Извикваме алгоритъма за въпросния файл и резултата от него записваме в друг файл. Клиентът превръща информацията от последния файл в удобен за четене вид и я изкарва на конзолата.

3. ПРОЕКТИРАНЕ

Клиентът ще изпрати всички свои данни, ред по ред към сървъра. Той ще ги запази във файл, пусне алгоритъма и след това клиентът ще прочете от файла.

Нужна ни е библиотеката **SPMF** - алгоритъм за frequent **itemsets**

Нужни са ни отделни сървър клас, клиент клас и клас, който ще създаде две нишки в които ще се изпълняват съответно клиентът и сървърът, като се извикат методите им.

4. РЕАЛИЗАЦИЯ

Създават се отделни класове **Server**, **Client** и **MainClass**. Клиент класа има 4 метода – **launchClient**, **addToList**, **encode** и **decode**. В **launchClient** се четат данните от входния файл ред по ред и се записват във списък чрез извикването на метода **addToList**. Данните от списъка се кодират в числен вид чрез метода encode, като се използва масив от цели числа, в който се записват числените стойности, като на еднаквите елементи от списъка съответстват еднакви елементи от масива. Отваря се сокет към хост ‘localhost’, порт 2019 и започват да му се прашат данните от масива елемент по елемент. Сървърът от своя страна има един метод – **launchServer**. В него се създава обект от тип **ServerSocket**, като в конструктора му се задава порт 2019, и обект от тип Socket, който accept-ва и установява връзката с нашият клиент. Сървърът започва да получава данните от масива и ги записва във файл с име temp.txt, като всеки елемент е на отделен ред (така подаваме на алгоритъма множества само с по един елемент). Извиква се алгоритъма, който приема: името на файла, в който сме записвали до сега, име на файл, в който ще се изведе резултата от изпълнението на алгоритъма и число от тип double (**minsup**), което указва на алгоритъма да покаже данните който се срещат в над minsup процента от всички. Сега клиентът в метода **launchClient** извиква метода **decode**, който чете информацията от файла, в който е записан резултатът от изпълнението на алгоритъма, превръща я в удобен за четене вид и я извежда на стандартния изход.

В главния метод на главния клас създаваме две нишки. В **run()** метода на едната извикваме метода **launchServer** от класа **Server**, а в **run()** метода на другата извикваме метода **launchClient** от класа **Client**.

5. РЪКОВОДСТВО ЗА ИНСТАЛАЦИЯ

В главната директория се създава файл, съдържащ стойности, разделени със запетая и данните се поставят в указания формат. Приложението се стартира с начална точка **MainClass.java**.

6. РЕЗУЛТАТИ

На стандартния изход на клиента се извеждат най-достъпваните ресурси от тип Event context.

7. ИЗПОЛЗВАНА ЛИТЕРАТУРА

* Създаване на клиент и сървър и свързването им: http://cs.lmu.edu/~ray/notes/javanetexamples/?fbclid=IwAR1rv50vyTG6a1oEZrEv05aJbAuJ8ZPBi5geO-5P1DHES4A3UfijWoXzVZk
* Работа с входни и изходни потоци: <http://www.ntu.edu.sg/home/ehchua/programming/java/j5b_io.html>
* Документация и алгоритми от SPMF:

<http://www.philippe-fournier-viger.com/spmf/>

Клиентът прочита резултата от алгоритъма от файла, превръща го в удобен за четене вид и го извежда на конзолата.

Сървърът извиква алгоритъмът за файла и записва резултата от алгоритъма в друг файл.

Сървърът прочита информацията от клиента и записва всеки елемент от масива на отделен ред във файл.

Обхождаме списъка и записваме данните от него в числен вид в масив от цели числа.

Изпращаме елемент по елемент масива на сървъра.

Клиентът чете от подадения файл и записва в списък информацията от него.