

Západočeská univerzita v Plzni

Fakulta aplikovaných věd

Katedra informatiky a výpočetní techniky

## **Semestrální práce z předmětu PPR**

(Zadání č. 9)

Martin Podval

[martin@m4rtin.com](mailto:martin@m4rtin.com)

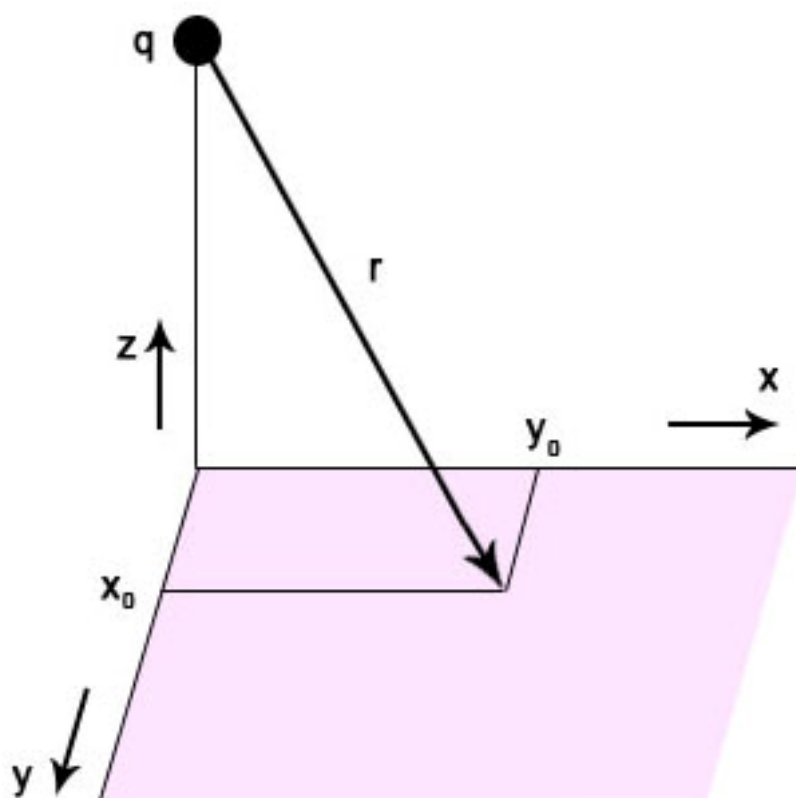
6.3.1981

## Zadání

Bodový elektrický náboj  $Q$  má v 3D prostoru souřadnice  $(0,0,z)$ . Realizujte výpočet toku intenzity elektrického pole  $E$  (vektor) danou obdélníkovou oblastí roviny  $(x,y)$ .

## Fyzikální řešení

V prostoru máme umístěn bodový náboj, který má hodnotu  $q$ . Dále máme obdelníkovou rovinu o rozměrech  $x$  krát  $y$ . Bodový náboj je od počátku roviny vzdálen o hodnotu  $z$ . V rovině je konstantně umístěn na souřadnici  $[0,0]$ , to znamená, že se posouvá pouze po  $z$ -tové souřadnici.



$$E = \frac{1}{4\pi\epsilon_0\epsilon_r} * \frac{|Q|}{r^2}$$

$$r = \sqrt{z^2 + \sqrt{x_0^2 + y_0^2}}$$

kde  $\epsilon = 4 * 10^{-7}$  je permeabilita prostředí. Podle výše uvedeného vzorečku tedy můžeme spočítat intenzitu v jakémkoliv bodě, stačí znát hodnotu náboje a vzdálenost od měřícího bodu.

## ***Výpočet na multiprocesoru se sdílenou pamětí***

Pro realizaci výpočtu se sdílenou pamětí byl použit programovací jazyk Java, který obsahuje silnou vhodnou podporu pro práci s vlákny.

### **Objektový rozbor úlohy**

Celý projekt byl rozdělen do 7 tříd, které spolu kooperují a jsou vzájemně propojeny.

#### **Vektor**

Jednoduchá třída, která obsahuje atributy a metody pro práci s dvou nebo tří rozměrnými vektory. Instance tohoto objektu jsou v celém projektu využívány pro uchování hodnot vektorů.

```
private int x;  
private int y;  
private int z;  
private boolean third;
```

#### **Náboj**

Třída obsahující nezbytné informace a metody pro definici náboje, tedy vektor jeho umístění a vlastní hodnotu náboj.

```
private Vektor souradnice;  
private double naboj;
```

#### **Plocha**

Jedna z hlavních tříd jejíž atributy popisují vlastnosti plochy, určené pro výpočet. Atributy této třídy jsou rozměry plochy, pole hodnot intenzity a pole vektorů intenzity pro celou plochu.

```
private int rozmerX;  
private int rozmerY;  
private double[][] intenzity;  
private Vektor[][] smery;
```

#### **Dělník**

Provádí výpočet vlákem. Z předaných parametrů, kterými jsou sloupce a jejich délky, vypočítá výsledné intenzity a ty zapíše pomocí funkce z instance třídy *Plocha* do výsledku.

```
private Model model;  
private boolean jeX;  
private ArrayList sloupce;  
private int pocet;
```

#### **Model**

Hlavní funkční třída, která obsahuje data celé modelové situace, nutná pro celý výpočet. Krom vlastností, která souvisejí s prostředím (permeabilita a počet procesů) jsou její atributy instance objektů náboje a plochy. V této třídě se vytváří a spouští instance vláken Dělníka a počte čas běhu výpočtu výsledných hodnot.

```
public final double epsilon = 4e-7;
private Naboj naboj;
private Plocha plocha;
private double epsilon_r;
private int pocetProcesu;
private long casBehu;
```

## Hlavní

Slouží pouze pro spuštění celého projektu vytvořením a provedením instance třídy *Model*.  
Též kontroluje zadané vstupní parametry a vypisuje nápovědu programu.

## TextExport

Poslední třída se zabývá exportem parametrů a výsledků do výstupního HTML souboru.  
Jediným atributem je instance objektu *Model*.

## Běh programu

Program se spouští s parametry, které jsou jednotlivé nastavitelné atributy nutné pro vyřešení celého problému výpočtu intenzit. Třída *Hlavní* obsahuje statickou metodu *main()*.  
Ta vytvoří instanci třídy *Model*. Do konstruktoru jsou poslány všechny atributy a na jejich základě jsou vytvořeny objekty náboje a plochy. Celý proces výpočtu se provede zavoláním metody *Model.vypocti()*. Ta vytvoří zadaný počet vláken dělníků, kteří počítají hodnoty intenzit. Třída *plocha* obsahuje synchronizovanou metodu pro zápis do pole intenzit a vektorů. Po ukončení běhu posledního z vláken dojde ke spočtení času běhu výpočtu. Jako poslední krok přijde na řadu vytvoření instance třídy *TextExport*, jenž uloží výsledná data do výstupního souboru.

## Spuštění programu

```
java -jar naboj.jar naboj_x naboj_y naboj_z hodnota_naboje plocha_x plocha_y
relativni_permeabilita pocet_vlaken
```

naboj_x, naboj_y, naboj_z	souřadnice náboje
hodnota	velikost náboje
plocha_x, plocha_y	rozměry plochy určené pro výpočet
relativni_permeabilita	hodnota relativní permeability
pocet_vlaken	udává počet vláken, které budou vytvořeny při běhu

Př: java -jar naboj.jar 0 0 5 1 1000 1000 2 5

Verze jazyku Java: 1.4.1\_02-b06

## Výstup programu do konzole

Program vypisuje v průběhu výpočtu informace o stavu vláken na do uživatelské konzole.

Př: Program vypočte intenzitu pro zadanou plochu, parametry prostředí a náboj.

Souřadnice náboje: [0, 0, 5]  
Hodnota náboje: 1C  
Velikost plochy: 1000x1000

Relativní permeabilita: 2  
Počet procesů: 5

Vlákno číslo 1 spuštěno  
Vlákno číslo 2 spuštěno  
Vlákno číslo 3 spuštěno  
Vlákno číslo 4 spuštěno  
Vlákno číslo 5 spuštěno  
Vlákno číslo 1 ukončeno  
Vlákno číslo 2 ukončeno  
Vlákno číslo 3 ukončeno  
Vlákno číslo 4 ukončeno  
Vlákno číslo 5 ukončeno

Čas běhu výpočtu: 1161

### **Výstup programu do souboru**

Výsledky výpočtu jsou pro přehlednost uloženy do výstupního html souboru. Ten je umístěn v adresáři výpisy a má následující strukturu:

#### **hodnotyYYYYMMDDHHMMSS.html**

kde sekce s velkými písmeny je tzv. časová známka (timestamp), tak aby byl soubor vždy identický.

Př: hodnoty20050107075810.html

### ***Výpočet na multiprocesoru s distribuovanou pamětí***

Pro realizaci výpočtu se distribuovanou pamětí byl použit programovací jazyk C s knihovnou PVM.

#### **Objektový rozbor úlohy**

Program funguje klasicky jako farmer-worker. Jeden proces farmáře funguje tak, že rozděljuje práci pro jednotlivé procesy workerů, které počítají vlastní výpočty. Workeri po vlastní práci zašlou výsledky zpět a farmář je skládá dohromady.

#### **Popis souborů**

Farmer.c – obsahuje program farmáře.

Worker.c – obsahuje program dělníků.

V programu farmáře je uživatel vyzván k zadání vstupních hodnot potřebných pro výpočet (velikost náboje, permeabilita, atd.). Po jejich vyplnění v cyklu projde vstupní zadané velikosti plochy a dělníků rozdělí práci.

Až dělník vypočte zadanou práci, vrátí výsledek. Pokud jsou ještě nějaká data, která by měla být spočtena, farmář opět dělníkovi přidělí práci. Po doběhu cyklu je nutné počkat na doručení všech výsledků.

Dělník naopak přijímá práci a vypočtené hodnoty zasílá farmáři, který mu práci zadal.

## **Spuštění programu**

Po spuštění farmáře je nutné vyplnit příslušné vstupní hodnoty. Po proběhnutí výpočtu dojde k vypsání výsledků na obrazovku a ukončení programu.

## **Závěr**

Část semestrální práce psaná v Javě byla psána bez předlohy. Naopak pro část pro výpočet na multiprocessoru s distribuovanou pamětí byly použity zdroje v podobě příkladů obdržených v rámci předmětu.

První část práce jsem odladil pod Windows na mém stroji. Vývoj nečinil žádný větší problém, práce byla naopak zajímavá a přínosná, díky pěkně vymyšlenému způsobu používání vláken v jazyce Java.

Část s PVM jsem ladil na školním stroji jumbo. Prvním problémem byl jazyk C, který je poměrně zbytečně záludnější než Java a místo vývoje funkčnosti programu musí naopak programátor spíše ladit jednoduché věci jako je práce s pamětí a další. Ovšem celý způsob PVM je sám o sobě velice zajímavý.

Celá semestrální práce je nutnou a pěknou zkušeností studenta s paralelním programováním. Díky její hojné využitelnosti v praxi je to, ale bohužel pouhé minimum, které by bylo nutné k lepšímu ozřejmění věci, které budeme muset znát při práci po studiu.