

Budapesti Műszaki és Gazdaságtudományi Egyetem

Villamosmérnöki és Informatikai Kar

Szélessávú Hírközlés és Villamosságtan Tanszék

Poros plazma kísérletek támogatása multiprocesszoros környezetben

DIPLOMATERVEZÉS 1. - BESZÁMOLÓ

Készítette

Bakró Nagy István

Konzulens

Hartmann Péter & Reichardt András

2014. április 26.

Tartalomjegyzék

Kivonat	4
1. A Poisson egyenlet bemutatása	5
1.1. A Poisson egyenlet	5
1.2. Az elektrosztatikus probléma	5
2. A Poisson egyenlet diszkretizálása és megoldási módszerei	6
2.1. A probléma definiálása	6
2.2. A háló illesztése	6
2.3. Deriváltak közelítése és az egyenlet megoldása	6
3. Nvidia CUDA platform ismertetése	7
3.1. CUDA-val szerelt GPU-k bemutatása	7
3.2. Architektúra és programozása	7
3.3. Mintaprogram: Két vektor összegzése blokkok és szálak használatával	7
4. Megoldók ismertetése és összehasonlítása	8
4.1. Megoldó MATLAB környezetben	8
4.2. Megoldó CUDA-C nyelven	8
4.3. Összehasonlítás	8
5. Összegzés	9
Ábrák jegyzéke	10
Táblázatok jegyzéke	11
Irodalomjegyzék	11

FELADATKIÍRÁS

A modern alacsony hőmérsékletű plazmafizikai kísérletek egy új, érdekes és izgalmas területe a poros plazmák kutatása. Egy elektromos gázkisülésbe helyezett apró (mikrométer méretű) szilárd szemcse a kisülési plazma atomi részecskéivel kölcsönhatva elektromosan feltöltődik. A sok töltött szemcséből kialakuló elrendezésben a szilárdtestfizikai jelenségek széles spektruma figyelhető meg, pl. kristályrács kialakulása, fázisátalakulás, diszlokációk dinamikája, transzport folyamatok, stb. Poros plazmákat jelenleg leginkább alapkutatásokban alkalmaznak, de jelentőségük az elektronikai gyártásban, fúziós reaktorok üzemeltetésében, Teraherz technológiában egyre inkább előtérbe kerül.

A kísérleti adadtgyűjtés és feldolgozás nagyrésze részecske-követő velocimetrián (particle tracking velocimetry) alapul, vagyis első lépésben egy nagysebességű kamera segítségével nagyfelbontású képek készülnek, amely képek segítségével a porszemcsék pontos (a kamera felbontásánál pontosabb) koordinátáit kell meghatározni. A képek elemzése ezidáig csak a mérést követően, hosszú idő alatt volt megvalósítható a vizsgálandó nagy adatmennyiség miatt. A multiprocesszoros környezetek segítségével a feldolgozás gyorsítása lehetséges akár több nagyságrenddel is.

A jelölt feladata, hogy a meglévő kísérleti elrendezés, amely az MTA Wigner Fizikai Kutatóközpont Szilárdtestfizikai és Optikai Intézetben található, kiegészítésével a mérés közbeni feldolgozással a mérést segítő analízist hajtson végre. Ennek eredményével a mérés előkészítése és elvégzése lényegesen gyorsulhatnak.

A jelölt feladata

- Mutassa be a mérési elrendezést és elemezze a kapott adatokat! (Mutassa be a mérést!)
- Elemezze a lehetséges multiprocesszoros környezeteket, a feladat szempontjából lényeges paraméterek és feladatvégrehajtási elvárások szempontjából!
- Készítsen programot, amely az azonnali (valós idejű) analízisben résztvevő paramétereket számítja ki, a multiprocesszoros környezet kihasználása nélkül!
- Készítsen programot, amely a mérési környezetbe illeszkedve a mérésnél valós időben képes a vizsgált paraméterek megjelenítésére! Mutassa be és elemezze az elkészített programot!
- Hasonlítsa össze a multiprocesszoros és a nem-multiprocesszoros környezetre elkészített programokat erőforrás igény illetve egyéb paraméterek szempontjából!

Tanszéki konzulens: Reichardt András, egy. tanársegéd

Külső konzulens: Hartmann Péter, PhD., tud. főmunkatárs (MTA Wigner FK, SZFI)

Budapest, 2014.03.10.

Kivonat

hablaty

1. fejezet

A Poisson egyenlet bemutatása

1.1. A Poisson egyenlet

1.2. Az elektrosztatikus probléma

2. fejezet

A Poisson egyenlet diszkretizálása és megoldási módszerei

2.1. A probléma definiálása

2.2. A háló illesztése

2.3. Deriváltak közelítése és az egyenlet megoldása

3. fejezet

Nvidia CUDA platform ismertetése

3.1. CUDA-val szerelt GPU-k bemutatása

3.2. Architektúra és programozása

3.3. Mintaprogram: Két vektor összegzése blokkok és szálak használatával

Az eszközön futó kód:

```
// Az eszkozon futo mag.
__global__ void add(int *a, int *b, int *c) {
    int index = threadIdx.x + blockIdx.x * blockDim.x;
    c[index] = a[index] + b[index];
}
```


4. fejezet

Megoldók ismertetése és összehasonlítása

4.1. Megoldó MATLAB környezetben

4.2. Megoldó CUDA-C nyelven

4.3. Összehasonlítás

5. fejezet

Összegzés

Dolgozatomban ...

Ábrák jegyzéke

Táblázatok jegyzéke