



УБРЗАВАЊЕ АПЛИКАЦИЈЕ КОЈА РЕШАВА ГРОС-ПИТАЕВСКИ ЈЕДНАЧИНУ КОРИШЋЕЊЕМ МАКСЕЛЕР МАШИНЕ ЗА ПРОТОЧНУ ОБРАДУ ПОДАТАКА

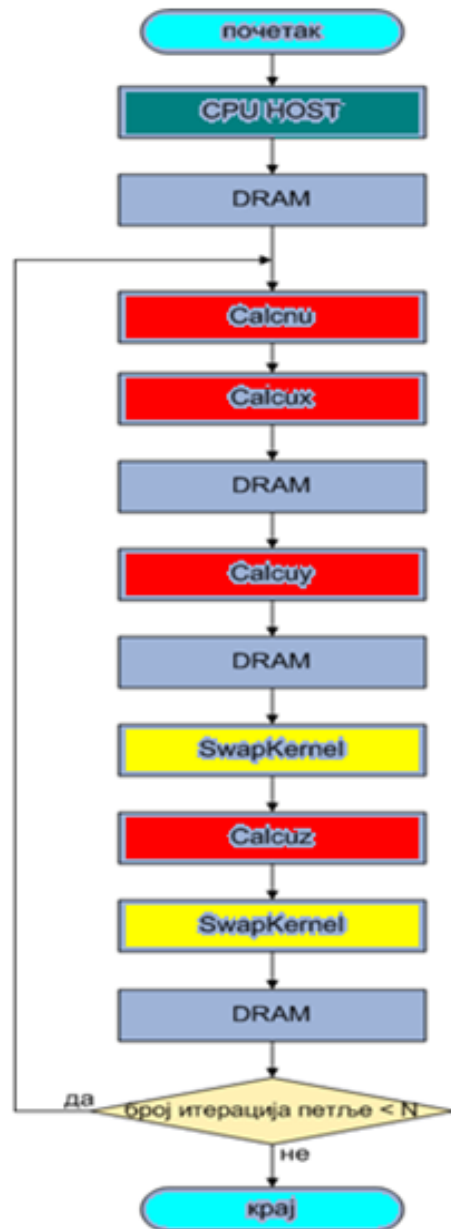
Лена Парезановић

**Ментор:
Проф. Др Вељко
Милутиновић**

Опис проблема

- Грос-Питаевски једначина
- Конкретан проблем
- Део кода који се убрзава

```
for(cnti = 0; cnti < Npas; cnti++) {  
    calcnu(psi);  
    calclux(psi, cbeta);  
    calcluy(psi, cbeta);  
    calcluz(psi, cbeta);  
}
```



Опис решења

- Типови података
- Приступ подацима у меморији
 - Акценат на максималном искоришћењу *pipeline-a*
- Имплементација функционалности
- Синхронизација процесора домаћина и машине за проточну обраду

Типови података

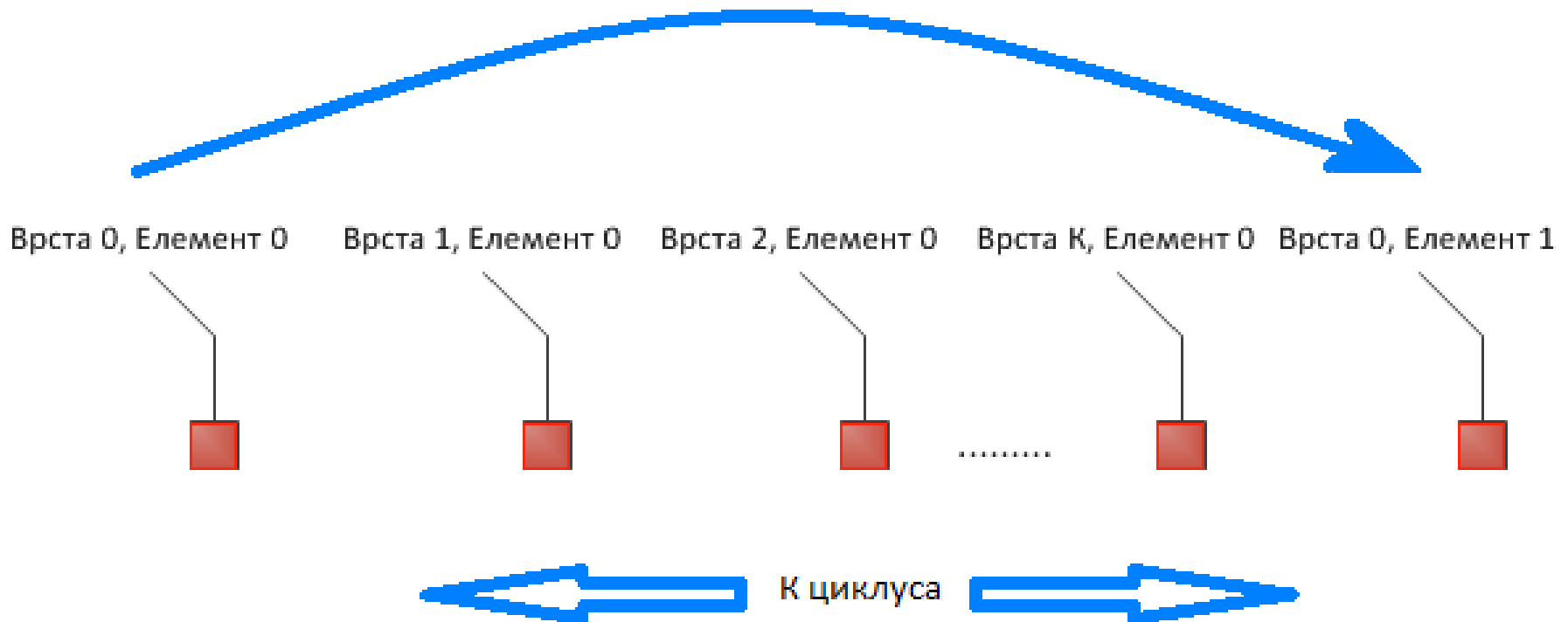
- Комплексни бројеви :
 - реални и имагинарни део
 - float point бројеви двоструке прецизности

```
KComplexType cplx_type = new KComplexType(hwFloat(11, 53));
```

```
HWFloat doubleType = hwFloat(11, 53);
```

Пристап подацима у меморији 1/2

- Основни циљ:
 - Максимално искоришћење pipeline-a



Пристап подацима у меморији 2/2

- Приступ по
 - x оси
 - y оси
 - z оси
- Генерисање командних токова
- Помоћни кернел
за промену редоследа података
учитаних по z оси

Имплементација функционалности 1/2

- Кернели за функције:

- 1) *calcpu*

- Једноставна имплементација
 - Секвенцијалан приступ меморији

- 2) *calclux, calclu, calcluz*

- Приступ подацима по одговарајућој оси у директном и инверзном редоследу
 - Потребно обртање кроз блоковски *RAM*

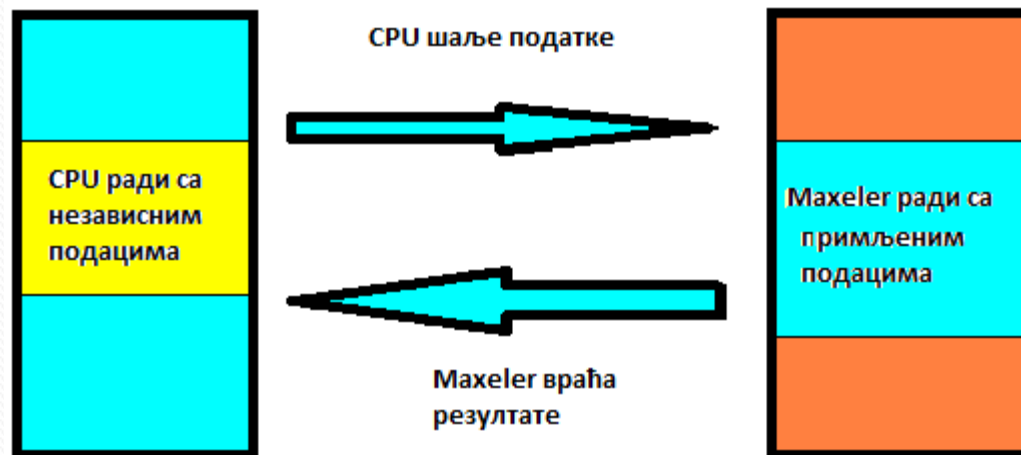
Имплементација функционалности 2/2

- Кернели за функције: *calclux*, *calclu*, *calcluz*
 - I група операција
 - приступ подацима у редоследу у којем су учитани
 - Обртање података
 - кроз блоковски RAM
 - II група операција
 - приступ подацима у инверзном редоследу у односу на који су учитани



Синхронизација процесора домаћина и машине за проточну обраду

- У паралели :
 - чекати податке
 - обављати операције независне од њих



Резултати

- Први случај:

3D низ $432 * 432 * 432$.

Број итерација три петље који се убрзавају: 60, 4, 1.

Време, *Intel i3*: 1356 секунди.

Време, Maxeler MAX2: 164 секунде.

- Други случај - реални параметри апликације:

3D низ димензија $432 * 432 * 432$.

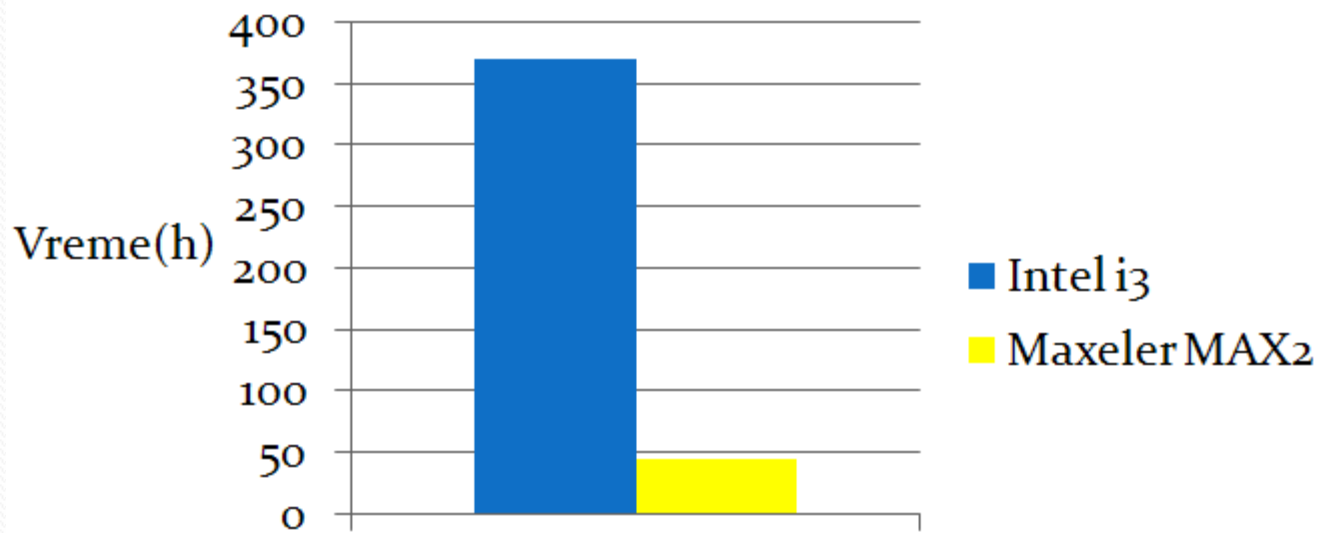
Број итерација три петље који се убрзавају:
60000, 4000, 1000.

Време, *Intel i3*: 371 сат.

Време, Maxeler MAX2: 45 сати.

Резултат

- У односу на процесор *intel i3*,
убрзање 8,24 пута
за исте параметре извршавања

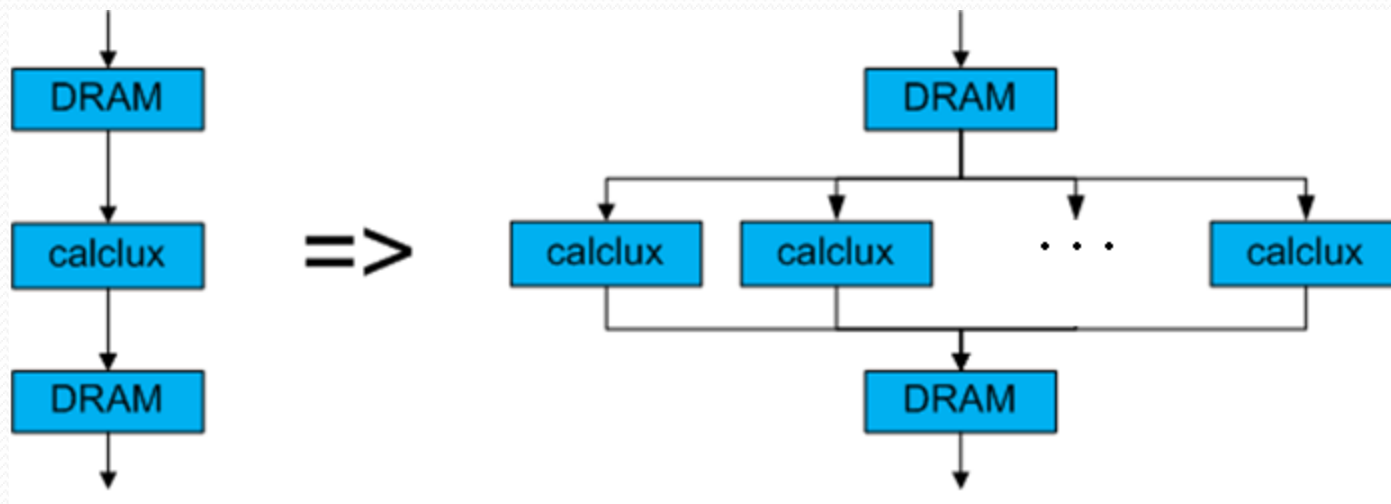


Потрошња енергије

- Intel i3
 - $175\text{W} * 371\text{h} = 64925 \text{ Wh}$
- Maxeler MAX2
 - $86\text{W} * 45\text{h} = 3870\text{Wh}$
- Однос
 - смањена потрошња 16.8 пута

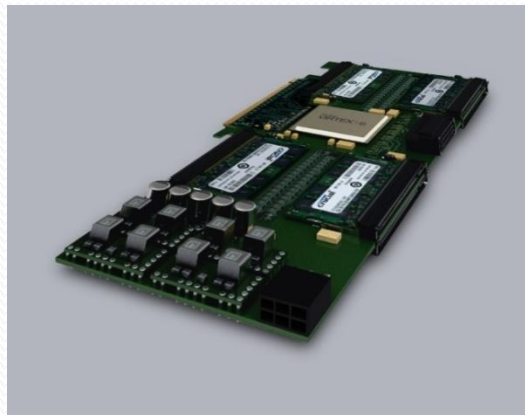
Даља побољшања

- Повећати паралелизам
мултиплицирањем кернела
спрам броја слободних ресурса



Закључак

- Научне апликације попут ове представљају идеалне кандидате за убрзавање помоћу Макселер машине за проточну обраду података



Acceleration of an algorithm based on the 3D Gross Pitaevskii equation

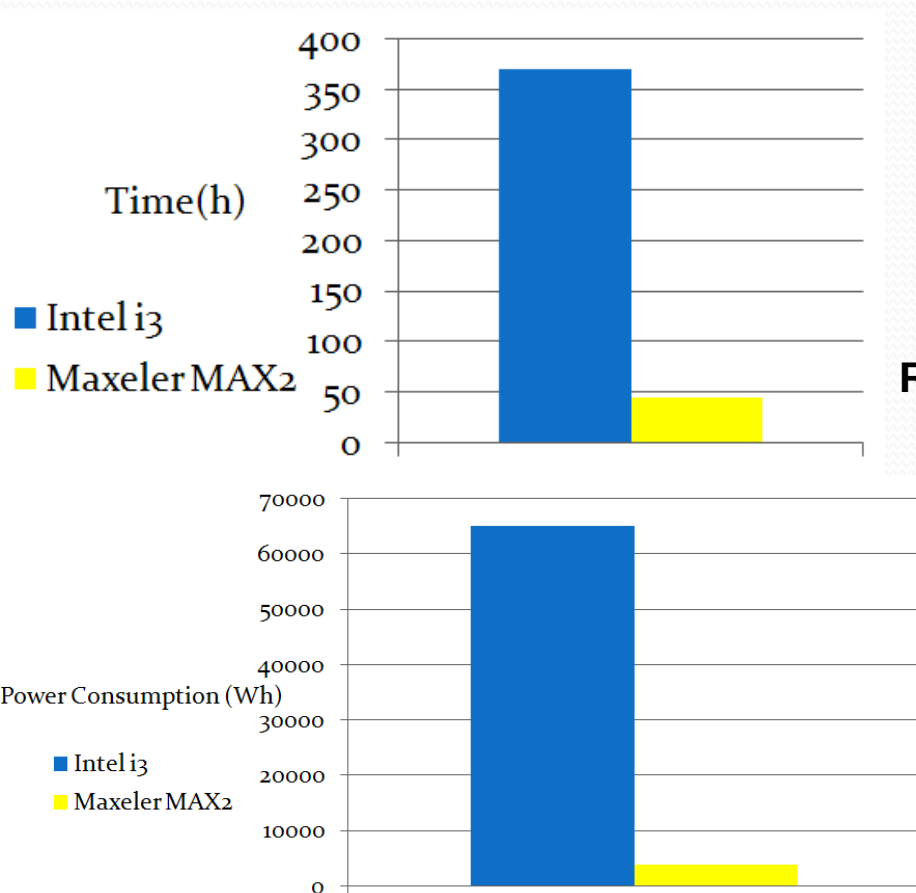
Execution time of the application:

CPU execution time: 371h

Dataflow execution time: 45h

Achieved speedup: 8.24x

Reduction of power consumption: 16.8x



Programmer:

Lena Parezanovic

lena_parezanovic@hotmail.com

Advisor:

Veljko Milutinovic

vm@etf.rs



University of
Belgrade

Note: Comparison is made between Intel i3 CPU and a single dataflow implementation on the oldest and smallest MAX 2 Maxeler card.