

Универзитет у Београду
Електротехнички факултет
Катедра за рачунарску технику и информатику

Списак могућих тема за завршне и мастер радове на предмету
Мултипроцесорски системи (13С114МУПС, 13Е114МУПС)
за студенте на Одсеку за РТИ и Одсеку за СИ

(ажурирано 18.04.2022.)

Контакт:

Доц. др Марко Мишић (marko.misic@etf.bg.ac.rs)

Финална тема се дефинише у разговору са предметним наставником на основу оквирних тема које су наведене у прилогу.

Студенти могу сами да предложе тему из области које се обрађују на предмету и сродних области у којима се примењују мултипроцесорски системи и паралелно програмирање, као што су вештачка интелигенција, машинско учење, анализа социјалних мрежа, нумеричке методе, примене у симулацијама за природне науке и сл.

Оквирне теме:

- 1. Паралелизација графовских алгоритама коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе**

Полазећи од доступних секвенцијалних имплементација основних алгоритама за обраду графова (обиласци, проналажење минималног обухватног стабла, најкраћих растојања и сл.) потребно је урадити њихово прилагођавање и паралелизацију коришћењем једне од понуђених технологија.

- 2. Паралелизација алгоритама за планарно распоређивање графова на централном и графичком процесору коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе**

Тема обухвата паралелизацију једног или више познатих алгоритама, као што су *Fruchterman-Reingold*, *Yifan-Hu*, *Harel-Koren Fast Multiscale*, *Kamada-Kawai* или *Force Atlas*. Могућност писања plugin-а за софтверски пакет Gephi или самосталног алата за визуелизација у 2D или 3D.

- 3. Оптимизација и паралелизација проблема Ератостеновог сита за одређивање простих бројева**

Ератостеново сито је познати метод за одређивање простих бројева у задатом опсегу (https://en.wikipedia.org/wiki/Sieve_of_Eratosthenes). Постоји већи број варијанти основног проблема, од којих су неки погодни за паралелизацију. Могућа симулација и визуелизација целог процеса.

4. Упоредна анализа програмских модела за паралелизацију директивама и модела вишег нивоа

Анализа карактеристичних комерцијалних и академских програмских модела попут OpenMP (и варијанти као што је JOMP), OpenACC или oneAPI (SYCL i DPC++) на примеру неколико секвенцијалних апликација. Могућност евалуације преводиоца отвореног кода као што је class порт LLVM преводиоца (<https://www.exascaleproject.org/highlight/clacc-an-open-source-openacc-compiler-and-source-code-translation-project/>) или преводиоца високих перформанси као што PGI преводилац.

5. Анализа нумеричке стабилности алгоритама и имплементација различитих модела за постизање репродукцибилности нумеричких резултата у паралелном окружењу

Познат је проблем нумеричке нестабилности и репродукцибилности алгоритама у паралелном окружењу. Анализа и имплементација шема за постизање стабилности и репродукцибилности, као што су дугачки акумулатор, аритметика мешане прецизности и сл.

6. Алгоритми за детекцију комуна (заједница) у социјалним мрежама и графовима

Детекција комуна у графовима и социјалним мрежама. Имплементација метода попут label propagation, Girvan-Newman алгорита, Clauser-Newman-Moore и сл. Могућа паралелизација неком од доступних метода.

7. Паралелизација алгоритама за одређивање елиптичних кривих и асиметричну криптографију коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе

Паралелизација алгоритама заснованих на ed25519 DSA систему за криптовање и SECP256K1 елиптичној криви. Могућности интеграције у SUPERCOP benchmark скуп.

8. Анализа технике за оптимизацију петљи у оквиру секвенцијалних и паралелних програма написаних коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе

Анализа и оптимизација секвенцијалних и паралелних програма написаних у одговарајућем моделу коришћењем познатих техника за оптимизацију петљи: *loop unrolling*, *loop fusion*, *loop fission*, *loop distribution*, *loop skewing*, *loop peeling*, *loop interchange* и сл. Примери техника се могу видети на: <https://johnysslabs.com/loop-optimizations-how-does-the-compiler-do-it/>, као и у књизи *Kennedy, Ken, and John R. Allen. Optimizing compilers for modern architectures: a dependence-based approach. Morgan Kaufmann Publishers Inc., 2001.*

9. Паралелизација проблема интеракције чврстих тела (*n-body*) коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе

Паралелизација проблема интеракције чврстих тела (*n-body*) коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе и неког од познатих метода, као што су Barnes-Hut, Fast Multipole Method и сл.

10. Превођење CUDA кода у портабилни C++ код коришћењем HIP алата

Анализа и трансформација паралелних програма написаних за CUDA платформу у портабилни C++ код коришћењем HIP алата чиме се омогућава портабилно извршавање на NVIDIA и AMD графичким процесорима. Анализа на примеру неколико доступних, карактеристичних CUDA програма.

11. Паралелизација алгоритама за детекцију шаблона, облика или ивица коришћењем OpenMP/CUDA/OpenACC/OpenCL платформе

Паралелизација и коришћење различитих алгоритама за проналажење шаблона, облика или ивица на сликама или у стринговима коришћењем једне од понуђених технологија. Коришћење алгоритама за упаривање стрингова као што су *Boyer-Moore*, *Aho-Corrasick*, *Karp-Rabin* и сл.

12. Имплементација паралелних апликација коришћењем готових библиотека или специфичних и нових програмских модела

Имплементација доступних, карактеристичних секвенцијалних апликација коришћењем библиотека као што су *ArrayFire*, *Julia*, *UPC*, *Chapel* програмски језици, *Kokkos* и *Raja* радни оквири и сл.

13. Тестирање централних и графичких процесора коришћењем микро-тестова

Испитивање различитих аспеката рада централног и графичког процесора попут величине кеш меморије, начина извршавања нити и сл. коришћењем наменских микро тестова и њихових комбинација. Испитивања меморијске архитектуре.

14. Примена централних или графичких процесора у обради звука

Програмирање самосталних апликација или додатака (*plugin*-ова) за програмски пакет као што је *Audacity*. Паралелизација ефеката попут појачавања и стишавања звука, уклањања еха, кашњења и сл. помоћу једне од доступних технологија.

15. Дебаговање, профјалирање и оптимизација паралелних OpenMP/CUDA/OpenACC/OpenCL апликација

Испитивање могућности алата за дебаговање, профјалирање и оптимизацију паралелног програмског кода на примеру доступних, карактеристичних benchmark апликација. Примери алата као што су *NVIDIA Parallel Nsight*, *Intel VTune Profiler*, *Valgrind*, *Scalasca*, *Vampire* и *Tau*.

16. Анализа софтверских пакета за тестирање и евалуацију суперрачунарских платформи

Анализа софтверских benchmark пакета *Linpack*, *HPCG* и *HPGMG*. Упоредна анализа примене на неколико доступних платформи. Утицај на TOP500 и GREEN500 листе суперрачунарских система.

17. Имплементација протокола за одржавање кохеренције кеш меморије као веб апликације

Имплементација и адаптација постојећих протокола за одржавање кохеренције кеш меморије (*WTI*, *WTI – write allocate*, *MSI*, *MESI*, *MOESI*, *Dragon*, *Firefly*) коришћењем VivioJS радног оквира (<https://www.scss.tcd.ie/Jeremy.Jones/VivioJS/vivio.htm>) или наменски развијеног симулационог окружења.