

НАСТАВНО-НАУЧНОМ ВЕЋУ

Предмет: Подобност теме и кандидаткиње Маје Вукасовић, мастер инжењера електротехнике и рачунарства, за израду докторске дисертације под називом „Побољшање перформанси програма коришћењем делимично контекстно осетљивих профила“.

Одлуком 409/25-2 бр. од 14.03.2023. године, именовани смо за чланове Комисије за оцену подобности теме и кандидаткиње Маје Вукасовић, за израду докторске дисертације и научне заснованости теме **„Побољшање перформанси програма коришћењем делимично контекстно осетљивих профила“** (енг. *Improving Program Performance with Partially Context-Sensitive Profiles*)

На основу материјала приложеног уз Захтев кандидата, Комисија подноси следећи

ИЗВЕШТАЈ

1. Подаци о кандидату

1.1. Биографски подаци

Маја Вукасовић је рођена 25.05.1993. године у Београду. Основну школу завршила је у Београду, као ђак генерације и носилац дипломе Вук Караџић. Након тога завршава Девету гимназију „Михаило Петровић – Алас“, такође у Београду као носилац Вукове дипломе. Током школовања освајала је награде на такмичењима из математике, физике, српског и енглеског језика.

На Електротехнички факултет уписала се школске 2012/2013. године на Одсеку за софтверско инжењерство. Дипломирала је 11.07.2016. године са просечном оценом 9.98. Дипломски рад под називом „Напредни генератор програмског кода за машине стања на језику UML“ одбранила је са оценом 10. Након основних студија уписала је мастер академске студије у школској 2016/2017. години. Мастер рад под називом „Симулатор векторског процесора са предикатским извршавањем у више трака“ одбранила је 2017. године са оценом 10. Просечна оцена завршених мастер академских студија је 10.00.

Докторске студије уписала је 2017. године на Електротехничком факултету у Београду на модулу Софтверско инжењерство. Положила је све испите са оценом 10 и остварила 120 ЕСПБ. У истраживачком раду усмерила се ка програмским преводиоцима, са посебним нагласком на оптимизације током преводиоца програма. Објавила је 9 радова на страним и домаћим конференцијама као и један рад у домаћем часопису. Учествовала је на међународном ISSES пројекту, четири пројекта Министарства науке, технолошког развоја и иновација, као и пројекту Иновационог фонда „CoCos.ai“.

Током студија радила је на праксама у фирмама *SOL* и *Kudos* и била укључена на неколико пројеката компаније *Nordeus*. Школске 2013/2014. године прикључила се демонстраторском тиму на Катедри за рачунарску технику и информатику, а нешто касније и на Катедри за примењену математику.

На Катедри за рачунарску технику и информатику је запослена од 2016. године, најпре као сарадник у настави, а потом као асистент. Тренутно је ангажована на шест предмета који се изводе на студијским програмима Софтверско инжењерство и Рачунарска техника и информатика.

Рецензирала је радове на конференцијама TELFOR и ETRAN. Од 2018. године ангажована је као истраживач у фирми *Oracle Labs*. Коаутор је три наставна материјала из области заштите података.

1.2. Сечено научно-истраживачко искуство

Маја Вукасовић је у свом истраживачком раду показала посебно интересовање за област програмских преводаца, са посебним нагласком на оптимизације током превођења програма.

Уписала је докторске студије школске 2017/2018. године на модулу Софтверско инжењерство и у досадашњем току докторских студија остварила 120 ЕСПБ. Све испите предвиђене студијским програмом положила је са просечном оценом 10,00. Списак положених испита дат је у следећој табели:

Назив предмета	Оцена	ЕСПБ	Датум полагања
Сензорске мреже	10	9	22.02.2021.
Мултипроцесорски системи	10	9	09.02.2018.
Одабрана поглавља из програмских преводаца	10	9	19.09.2018.
Развој микропроцесорског софтвера	10	9	06.07.2021.
Сигурност и заштита рачунарских система	10	9	18.01.2018.
Увод у научни рад	10	6	24.09.2021.
Интернет програмирање	10	9	28.09.2020.
Мултимедијални системи – одабрана поглавља	10	9	13.02.2019.
Принципи програмских језика	10	9	24.09.2021.
Вештачка интелигенција и експертски системи	10	9	08.02.2020.

Кандидаткиња је испунила и све обавезе предвиђене наставним планом и програмом докторских студија: Студијски истраживачки рад I (15 ЕСПБ), Студијски истраживачки рад II (15 ЕСПБ) и Научно стручни рад (3 ЕСПБ).

Досадашњи резултати кандидаткиње приказани су кроз више публикација у домаћим и међународним часописима и међународним конференцијама. У даљем тексту наведен је списак публикација на којима је Маја Вукасовић коаутор:

Радови у домаћим часописима (M50):

1. Vukasović M., Veselinović B., Stanisavljević Ž., *Design and Implementation of a Configurable System for Managing X509 Certificates*, TELFOR JOURNAL, ISSN 1821-3251, Vol. 10, No. 2, pp. 102-107, Dec 2018, (M52), DOI: 10.5937/telfor1802102V

Радови на међународним конференцијама (M30):

1. Stanisavljević Ž., Walter B., Vukasović M., Todosijević A., Labedzki M., Wolski M., *GEANT Software Maturity Model*, TELFOR 2018, Belgrade, Electronic ISBN 978-1-5090-0054-8, pp. 691-694, Nov 2018, (M33), DOI: 10.1109/TELFOR.2018.8611887
2. Vukasović M., Miladinović D., Milaković A., Vuletić P., Stanisavljević Ž., *Programming Applications Suitable for Secure Multiparty Computation Based on Trusted Execution Environments*, TELFOR 2022, pp. 1-4, Belgrade, Nov, 2022.

Радови на домаћим конференцијама (M60):

1. Шекуларац Т., Хацић Ф., Тубић С., Вукасовић М., *Софтвер за учење и тестирање SQL језика*, Зборник радова 28. научне конференције "YU INFO 2022", Информационо друштво Србије, Копаоник, Србија, март 2022.
2. Кнежевић Б., Вукасовић М., Драшковић Д., *Решавање проблема проналажења потпуног подграфа*, Зборник радова 27. конференције "YU INFO 2021", pp. 42-47, Информационо друштво Србије, Копаоник, Србија, март 2021.
3. Драшковић Д., Цинцовић Ј., Мијаиловић Д., Вукасовић М., Јоцовић В., Милаковић А., *Предвиђање успеха студената студијског програма Софтверско инжењерство техникама машинског учења*, Зборник радова конференције "ЈУ ИНФО 2020", Информационо друштво Србије, Копаоник, Србија, март 2020
4. Вукасовић М., Веселиновић Б., Станисављевић Ж., *Развој конфигурабилног система за рад са X509 сертификатима*, ТЕЛФОР 2017, pp. 876-879, Београд, новембар 2017.
5. Delčev S., Vukasović M., Drašković D., Radojević D., Janković M., Bajec M., Nikolić B., *Testing artificial intelligence knowledge with interactive web system*, 24th Telecommunications Forum "TELFOR 2016", IEEE Serbia & Montenegro, Београд, Србија, Nov, 2016.
DOI: 10.1109/TELFOR.2016.7818922
6. Вукасовић М., Мићовић М., Раденковић У., Јоцовић В., Драшковић Д., *Примена виртуелне стварности у оквиру медицинских третмана*, 23. конференција "ЈУ ИНФО 2017", pp. 105-110, Информационо друштво Србије, Копаоник, Србија, март 2017.
ISBN: 978-86-85525-20-9
7. Драшковић Д., Мићовић М., Раденковић У., Вукасовић М., Делчев С., Јоцовић В., Шуштран Ж., Николић Б., *Реализација веб система за управљање и надгледање софтверских пројеката*, 23. конференција "ЈУ ИНФО 2017", pp. 99-104, Информационо друштво Србије, Копаоник, Србија, март 2017. ISBN: 978-86-85525-20-9

Дана 29. марта 2023. године, кандидаткиња Маја Вукасовић је на Електротехничком факултету Универзитета у Београду усмено полагала испит о подобности предложене теме докторске дисертације и кандидата (докторски испит), пред комисијом у саставу:

- др Мило Томашевић, редовни професор, Универзитет у Београду – Електротехнички факултет,
- др Милена Вујошевић Јаничић, ванредни професор, Универзитет у Београду – Математички факултет,
- др Коста Јовановић, ванредни професор, Универзитет у Београду – Електротехнички факултет.

На јавној усменој одбрани предложене теме Комисија је кандидаткињу оценила оценом задовољила.

1.3. Оцена подобности кандидата за рад на предложеној теми

Комисија је размотрила све претходно наведене резултате досадашњег рада кандидаткиње Маје Вукасовић, узимајући у обзир положене испите и стручне обавезе током докторских студија, успешну одбрану теме докторске дисертације и до сада објављене научне радове у часописима и на научно-стручним скуповима, чиме је кандидаткиња показала да је стекла неопходно искуство за самосталан научно-истраживачки рад. Комисија сматра да је кандидаткиња способна да током израде докторске дисертације оствари резултате, који могу бити приказани и верификовани од стране међународне научне заједнице. Комисија је утврдила да кандидаткиња Маја Вукасовић испуњава све потребне и формалне услове да приступи изради докторске дисертације на предложеној теми.

2. Предмет и циљ истраживања

Компилација, тј. превођење изворног програмског кода у извршни код се, у контексту програмског језика Јава и других језика који се заснивају на извршавању коришћењем виртуелних машина, назива превођење пре времена извршавања (енг. *ahead-of-time compilation*, скр. *AOT*). Током превођења програма спроводи се низ оптимизација како би генерисани извршни код био што ефикаснији. Многе оптимизације попут дупликације путева (енг. *path-duplication*), девиртуелизације полиморфних позива, алокације регистара (енг. *register allocation*), издвајања функција (енг. *function outlining*), уграђивања тела функција (енг. *function inlining*) [1, 2] као и бројних оптимизација петљи ослањају се на информације о извршавању програма који се преводи. Како су ове информације доступне тек у току извршавања програма, током АОТ превођења могу се користити искључиво информације о претходним извршавањима за исти или сличан програм [3, 4, 5]. Ове информације се називају профили, а добијене су процесом који се назива профилисање (енг. *profiling*). Метрички подаци који се налазе у профилима повезани су са контекстом у програмском коду, а контекст подразумева низ од једне или више локација у коду које се налазе на стеку позива функција у тренутку када је профил прикупљен. У зависности од броја локација у овом низу, контекст може бити мање или више прецизан, па се према овом критеријуму профили могу категоризовати као контекстно осетљиви или неосетљиви. У случају контекстно неосетљивих профила, контекст садржи искључиво информацију о позицији у оквиру функције са врха програмског стека. Контекстно осетљиви профили у општем случају садрже информације о већем броју локација на стеку узимајући у обзир и позиваоце функције са врха стека. Коришћењем контекстно осетљивих профила могуће је прецизније корелисати метрике са стварном локацијом у програму.

Предмет рада је детекција често извршаваних делова кода (енг. *hot-code detection*), измена распореда превођења (енг. *compilation scheduling*) и оптимизација инлајновања функција, са посебним нагласком на употребу профила ради побољшања одлука које се доносе у оквиру ове оптимизације. Већина преводилаца ослања се на оптимизације у оквиру функција (*intraprocedural optimizations*) [6, 7, 8] што представља мотивацију за стављање фокуса истраживања на ове области. Оптимизација инлајновања подразумева уграђивање тела позване функције на место позива и тиме елиминише трошкове самог позива, а омогућава оптимизовање уграђеног кода на основу локације уграђивања. Осим тога, ова оптимизација директно повећава тела функција у којима је дошло до уграђивања кода и, самим тим, отвара могућност за више оптимизација. Са друге стране превелико коришћење ове оптимизације доводи до повећања генерисаног кода, па се мора спроводити контролисано. Значајан број различитих хеуристичких решења оптимизације инлајновања спроводи анализу на основу процењене цене и добитка (енг. *cost-benefit analysis*) приликом сваке замене позива функције њеним телом [1, 2, 9, 10, 11, 12, 13, 14]. Неки инлајнери се ослањају на статичке информације приликом доношења ових одлука, али, ипак, већина модерних инлајнера ослања се на динамички добијене податке попут фреквенције

извршавања делова кода или динамичког разрешавања типова за виртуелне позиве функција. Детекција често извршаваног кода је од посебног значаја за исправну употребу буџета оптимизација јер би, у супротном, оптимизациони напори били уложени у код који се не извршава често, па би режијски трошкови спровођења оптимизација превазилазили корист добијену њиховом применом.

Циљ рада је развој и имплементација алгоритма за AOT преводилац који на основу делимично контекстно осетљивих профила омогућава детекцију и бољу оптимизацију често извршаваних делова кода без значајног увећања генерисаног кода. Да би ово постигао, потребно је да алгоритам на погодан начин измени редослед превођења и одлуке које се доносе током оптимизације инлајновања. Нагласак рада ће бити на имплементацији развијеног алгоритма за преводилац *Native Image* у оквиру *GraalVM* екосистема уз могућност прилагођавања другим преводиоцима који раде по истој парадигми и над истим или сличним форматом улазних података.

У оквиру истраживања биће коришћени делимично контекстно осетљиви профили. Њихова употреба представља један од начина за постизање компромиса између веће прецизности профила и цене процеса профилисања [8, 15, 16, 17, 18]. Овакви профили могу се ефикасно прикупити у оквиру *GraalVM* екосистема директно на основу графовске међурепрезентације преводиоца. Евалуација резултата имплементираних алгоритма биће спроведена над тестовима (енг. *benchmarks*) из скупова *Renaissance* [19], *DaCapo* [20] и *Scalabench* [21].

Значај истраживања огледа се у директним практичним применама решења које ће бити развијено и имплементирано. Наиме, време извршавања програма сматра се једним од основних показатеља перформанси програма и увек је од ултимативне важности да се оно што више смањи. Са друге стране, важно је и да примена оптимизација не узрокује претерано повећање генерисаног кода имајући у виду да је и меморија критичан ресурс у многим системима. Унапређење перформанси програма преведених пре извршавања применом нове технике превођења, која модификује редослед превођења и користи делимично контекстно осетљиве профиле да би потпомогла оптимизацију инлајновања, значајно је јер ће унапредити перформансе програма уз занемарљиво увећање генерисаног кода. О значају теме говори и чињеница да је ово веома активна и конкуритивна област и да постоје бројна истраживања са истим циљем о чему сведочи и велики број референци прикупљених у припреми ове теме.

Неке од најрелевантнијих референци из отворене литературе у вези са тезом су:

- [1] M. Arnold, S. Fink, V. Sarkar, and P. F. Sweeney. 2000. A Comparative Study of Static and Profile-based Heuristics for Inlining. In Proc. of the ACM SIGPLAN Workshop on Dynamic and Adaptive Compilation and Optimization. ACM, USA, 52–64. <https://doi.org/10.1145/351397.351416>
- [2] P. P. Chang, S. A. Mahlke, W. Y. Chen, and W. W. Hwu. 1992. Profile-guided Automatic Inline Expansion for C Programs. *Softw. Pract. Exper.* 22, 5, 349–369. <https://doi.org/10.1002/spe.4380220502>
- [3] E. Duesterwald and V. Bala. 2000. Software Profiling for Hot Path Prediction: Less is More. In Proc. of the Ninth International Conference on Architectural Support for Programming Languages and Operating Systems (ASPLOS IX). ACM, USA, 202–211. <https://doi.org/10.1145/378993.379241>
- [4] C. Krintz. 2003. Coupling on-line and off-line profile information to improve program performance. In Proc. of CGO 2003. 69–78. <https://doi.org/10.1109/CGO.2003.1191534>
- [5] D. Shabalin and M. Odersky. 2018. Interflow: Interprocedural Flow-Sensitive Type Inference and Method Duplication. In Proc. of the 9th ACM SIGPLAN International Symposium on Scala. ACM, USA, 61–71. <https://doi.org/10.1145/3241653.3241660>
- [6] G. Duboscq, T. Würthinger, L. Stadler, C. Wimmer, D. Simon, and H. Mössenböck. 2013. An

Intermediate Representation for Speculative Optimizations in a Dynamic Compiler. In Proc. of the 7th ACM Workshop on Virtual Machines and Intermediate Languages (VMIL '13). ACM, USA, 1–10. <https://doi.org/10.1145/2542142.2542143>

[7] M. Paleczny, C. Vick, and C. Click. 2001. The Java HotSpot Server Compiler. In Proc. of the 2001 Symposium on Java™ Virtual Machine Research and Technology Symposium - Vol. 1 (JVM'01). USENIX, USA, 1–1. <http://dl.acm.org/citation.cfm?id=1267847.1267848>

[8] C. Wimmer, C. Stancu, P. Hofer, V. Jovanovic, P. Wögerer, P. B. Kessler, O. Pliss, and T. Würthinger. 2019. Initialize Once, Start Fast: Application Initialization at Build Time. Proc. ACM Program. Lang. 3, OOPSLA, Article 184, 29 pages. <https://doi.org/10.1145/3360610>

[9] D. R. Chakrabarti and S-M. Liu. 2006. Inline Analysis: Beyond Selection Heuristics. In Proc. of CGO '06. IEEE Computer Society, USA, 221–232. <https://doi.org/10.1109/CGO.2006.17>

[10] P. P. Chang and W.-W. Hwu. 1989. Inline Function Expansion for Compiling C Programs. SIGPLAN Not. 24, 7, 246–257. <https://doi.org/10.1145/74818.74840>

[11] A. Prokopec, G. Duboscq, D. Leopoldseder, and T. Würthinger. 2019. An Optimization-driven Incremental Inline Substitution Algorithm for Just-in-time Compilers. In Proc. of CGO. IEEE Press, USA, 164–179. <http://dl.acm.org/citation.cfm?id=3314872.3314893>

[12] R. W. Scheifler. 1977. An Analysis of Inline Substitution for a Structured Programming Language. Commun. ACM 20, 9, 647–654. <https://doi.org/10.1145/359810.359830>

[13] M. Serrano. 1997. Inline expansion: When and how? Springer, 143–157. <https://doi.org/10.1007/BFb0033842>

[14] P. Zhao and J. N. Amaral. 2004. To Inline or Not to Inline? Enhanced Inlining Decisions. Springer 405–419. https://doi.org/10.1007/978-3-540-24644-2_26

[15] G. Ammons, J-D. Choi, M. Gupta, and N. Swamy. 2004. Finding and removing performance bottlenecks in large systems. In European Conference on Object-Oriented Programming. Springer, 172–196.

[16] G. Ausiello, C. Demetrescu, I. Finocchi, and D. Firmani. 2012. K-Calling Context Profiling. SIGPLAN Not. 47, 10, 867–878. <https://doi.org/10.1145/2398857.2384679>

[17] S. P. Reiss and M. Renieris. 2001. Encoding program executions. In Proc. of the 23rd International Conference on Software Engineering. ICSE 2001. IEEE, 221–230.

[18] M. Serrano and X. Zhuang. 2009. Building Approximate Calling Context from Partial Call Traces. In 2009 International Symposium on CGO. 221–230. <https://doi.org/10.1109/CGO.2009.12>

[19] A. Prokopec, A. Rosà, D. Leopoldseder, G. Duboscq, P. Tuma, M. Studener, L. Bulej, Y. Zheng, A. Villazón, D. Simon, T. Würthinger, and W. Binder. 2019. Renaissance: benchmarking suite for parallel applications on the JVM. In PLDI, ACM, 31–47. <http://dblp.uni-trier.de/db/conf/pldi/pldi2019.html#ProkopecRLD0SBZ19>

[20] S. M. Blackburn, R. Garner, C. Hoffmann, A. M. Khang, K. S. McKinley, R. Bentzur, A. Diwan, D. Feinberg, D. Frampton, S. Z. Guyer, M. Hirzel, A. Hosking, M. Jump, H. Lee, J. Eliot B. Moss, A. Phansalkar, D. Stefanović, T. VanDrunen, D. von Dincklage, and B. Wiedermann. 2006. The DaCapo Benchmarks: Java Benchmarking Development and Analysis. SIGPLAN Not. 41, 10, 169–190

[21] A. Sewe, M. Mezini, A. Sarimbekov, and W.r Binder. 2011. Da Capo Con Scala: Design and Analysis of a Scala Benchmark Suite for the Java Virtual Machine. In OOPSLA. 657–676.

3. Полезные гипотезы

Полазне хипотезе овог истраживања заснивају се на вишегодишњем искуству кандидатке у раду са продукционим екосистемом *GraalVM*. Основне хипотезе су:

- Потребни профили се могу ефикасно прикупити извршавањем наменски преведеног извршног кода.
- Профили садрже информације о фреквенцији извршавања делова кода који су идентификовани низом довољне дужине локација са програмског стека забележених у моменту прикупљања профила.
- Иако су повезани контексти парцијални и не садрже комплетну информацију о позицији у коду, захваљујући техници прикупљања профила метрике садржане у профилима су прецизне.
- На основу потпуних контекста и повезаних метрика у профилима могуће је прецизно одредити фреквенцију било којег позива неке функције у коду.
- Делимично контекстно осетљиве профиле је могуће апроксимативно експандовати тако да израчунате метрике буду приближно једнаке вредностима које би стварно биле измерене у случају прикупљања дужих контекста.
- Експандоване делимично контекстно осетљиве профиле је могуће груписати и филтрирати тако да апроксимирају групе често извршаваних делова кода.
- Могуће је променити редослед превођења тако да се да предност деловима кода који се често извршавају.
- Могуће је изменити алгоритам за примену оптимизације инлајновања тако да се фаворизују апроксимирани контексти који “моделују” често извршаване делове кода

4. Научне методе истраживања

У овом раду је придвеђена примена више истраживачких метода. На почетку истраживања биће извршена детаљна евалуација постојеће шеме превођења са посебним нагласком на претходно имплементирану оптимизацију инлајновања у оквиру преводиоца *GraalVM Native Image*. Како ова оптимизација подразумева алгоритам који зависи од значајног броја параметара, потребно је најпре квантификовати утицај појединачних параметара на перформансе програма који се преводи, а потом показати да се перформансе програма не могу значајно унапредити искључиво варијацијом вредности ових параметара.

Такође, на скупу стандардних бенчмарка ће бити извршена евалуациона анализа перформанси неколико релевантних конфигурација са којима треба упоредити резултат овог истраживања. Прве две конфигурације од интереса обухватају АОТ превођење *GraalVM Native Image* преводиоцем без употребе профила и са њима. Конфигурација која се ослања на профиле током превођења користиће претходно имплементиран алгоритам за укључивање профила у постојеће оптимизације и она ће бити најзначајнија за утврђивање доприноса новог алгоритма. Трећа конфигурација треба да представља превођење програма у току извршавања (енг. *just-in-time*, скр. *JIT*) употребом *Graal* преводиоца на *HotSpot* виртуалној машини. Очекује се да ова конфигурација прикаже најбоље перформансе због прецизности информација о програму који се преводи и механизма који не постоје у АОТ преводиоцу, а то су управо перформансе којима ново решење треба да тежи.

Улаз новог алгоритма је директно одређен међурепрезентацијом преводиоца и представља делимично контекстно осетљиве профиле. Посебан сегмент анализе постојећих решења

фокусираће се на методе прикупљања профила овог формата и њихове употребе да би се детектовали делови кода који се често извршавају. Такође, биће истражене и неке од постојећих хеуристика које користе профиле, имплементираних у различитим оптимизацијама са фокусом на оптимизацији инлајновања. Ипак, начин прикупљања профила у задатом окружењу диктира начин агрегирања и интерпретирања профила, што треба да омогући конструкцију специфичног решења.

5. Очекивани научни допринос

Очекује се да доприноси овог рада буду остварени у неколико праваца:

- Преглед и анализа постојећих решења употребе профила у оптимизацијама преводиоца са посебним нагласком на оптимизацију инлајновања,
- Преглед претходно предложених решења прикупљања и употребе делимично контекстно осетљивих профила,
- Опсежна класификација алата за профилисање,
- Предлог алгорита за детекцију често извршаваног кода на основу делимично контекстно осетљивих профила,
- Предлог алгорита за апроксимацију метрика дужих контекста на основу парцијалних контекста,
- Предлог измене редоследа превођења који даје првенство често извршаваним деловима кода,
- Предлог измене оптимизације инлајновања тако да користи хеуристику за додавање и квантификовање смерница о често извршаваном коду у *cost-benefit* анализу и, сходно томе, модификује буџет оптимизације,
- Имплементација предложених алгорита у продукционој компајлерској инфраструктури (*GraalVM* екосистем),
- Опсежна евалуација имплементираних алгорита и релевантних постојећих делова инфраструктуре над репрезентативним скупом бенчмарка,
- Поређење добијених резултата са резултатима релевантних истраживања.

6. План истраживања и структура рада

Истраживање би требало спровести у следећим фазама:

- Истраживање ће започети детаљном евалуацијом релевантних делова постојеће инфраструктуре *GraalVM*, који се односе, пре свега, на утврђени редослед и шему АОТ превођења, употребу профила у постојећим оптимизацијама са нагласком на оптимизацију инлајновања, као и перформансе ЈИТ преводиоца. На основу ове евалуације биће анализирани недостаци постојећег решења, као сагледани резултати којима треба тежити у оквиру овог истраживања.
- Затим ће бити спроведена исцрпна анализа постојећих решења у областима од интереса попут детекције значајних делова кода, употребе парцијалних профила за њихово означавање и употребе профила у циљу побољшања АОТ превођења и оптимизација.
- Главни део истраживања представљаће конструкција алгорита за модификацију редоследа и начина превођења програма у АОТ преводиоцу на основу информација о често извршаваним деловима кода, који се детектују употребом делимично контекстно осетљивих профила. Циљ алгорита је да повећа брзину извршног кода и да том приликом не повећа значајно величину генерисаног кода.

- Идејно решење ће бити имплементирано у оквиру преводиоца *GraalVM Native Image AOT*.

У завршној фази биће извршена евалуација добијених резултата над скупом бенчмарка. На основу ових резултата биће могуће извући закључке о успешности истраживања и смерницама за даље унапређивање рада.

7. Закључак и предлог

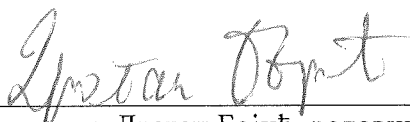
На основу приложене документације и стечене слике о теми докторске дисертације и досадашњем научном раду Маје Вукасовић, мастер инжењера електротехнике и рачунарства, Комисија закључује следеће:

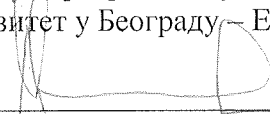
1. Кандидаткиња испуњава све формалне и суштинске услове прописане Законом о високом образовању, Статутом Универзитета у Београду и Статутом Електротехничког факултета Универзитета у Београду за израду докторске дисертације.
2. Предложена тема „Побољшање перформанси програма коришћењем делимично контекстно осетљивих профила“ је научно заснована, а резултати до којих ће се доћи израдом ове дисертације представљаће значајан и оригиналан допринос. Тема припада ужој научној области Софтверско инжењерство.

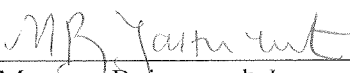
На основу претходно изложеног, Комисија са задовољством предлаже Наставно-научном већу Електротехничког факултета Универзитета у Београду да кандидаткињи Маји Вукасовић одобри израду докторске тезе под насловом „Побољшање перформанси програма коришћењем делимично контекстно осетљивих профила“ (енг. *Improving Program Performance with Partially Context-Sensitive Profiles*), а под менторством др Драгана Бојића, редовног професора Електротехничког факултета у Београду.

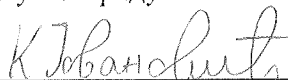
У Београду 30.03.2023. године

ЧЛАНОВИ КОМИСИЈЕ


др Драган Бојић, редовни професор
Универзитет у Београду – Електротехнички факултет


др Мило Томашевић, редовни професор
Универзитет у Београду – Електротехнички факултет


др Милена Вујошевић Јаничић, ванредни професор,
Универзитет у Београду – Математички факултет


др Коста Јовановић, ванредни професор
Универзитет у Београду – Електротехнички факултет