

Ekstrakcija informacija o prometnim zagušenjima iz velikih baza podataka GPS-om praćenih vozila

Tušek, Roman

Master's thesis / Diplomski rad

2016

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:017735>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-12-29**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences - Institutional Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Roman Tušek

**EKSTRAKCIJA INFORMACIJA O PROMETNIM ZAGUŠENJIMA IZ
VELIKIH BAZA PODATAKA GPS-OM PRAĆENIH VOZILA**

DIPLOMSKI RAD

Zagreb, 2016.

Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

EKSTRAKCIJA INFORMACIJA O PROMETNIM ZAGUŠENJIMA IZ VELIKIH BAZA PODATAKA GPS-OM PRAĆENIH VOZILA EXTRACTION OF TRAFFIC CONGESTION INFORMATION FROM LARGE DATABASES OF GPS TRACKED VEHICLES

Mentor: izv. prof. dr. sc. Tonči Carić
Roman Tušek

Student:
JMBAG: 0135213780

Zagreb, srpanj 2016.

Zahvala:

Zahvaljujem se svome mentoru i voditelju moga rada izv. prof. dr. sc. Tončiju Cariću za svo strpljenje i trud prilikom izrade ovog diplomskog rada. Svojim stručnim savjetima i sugestijama oblikovao je ideju koja je postala koncept ovog diplomskog rada. Također se zahvaljujem i asistentu Tomislavu Erdeliću, mag. ing. el. techn inf. na svojoj tehničkoj pomoći i iskustvu prilikom ustupljivanja arhivskih podataka

Posebno se želim zahvaliti svojoj majci i baki koje su me uvijek i bez sumnje podupirale, moralno i financijski, prilikom cijelog mog obrazovanja jer zbog njih sam dospio ovako daleko. Hvala vam na ukazanom povjerenju i ljubavi.

Želim se zahvaliti svojim kolegama čiji timski rad je uvelike utjecao na naše iskustvo na Fakultetu, svojim prijateljima na pomoći i savjetima te Moniki na svojoj ljubavi i podršci za vrijeme mog studiranja, tijekom najljepših i najtužnijih trenutaka.

Ekstrakcija informacija o prometnim zagušenjima iz velikih baza podataka GPS- om praćenih vozila

Sažetak: Ovaj diplomski rad opisuje jedan od načina pohranjivanja i obrade velikog broja profila brzina zasnovanih na GPS podacima unutar aplikacijskog rješenja te mjerenja učinkovitosti istog. Opisuje se izrada specijalizirane aplikacije koja obrađuje i vizualno prikazuje dane profile brzina. Opisani su unutarnji sistemi obrade, primjer korisničkog korištenja aplikacije te značenje profila brzina. Nakon toga izvršava se serija testova; spominju se testni programi, testno okruženje te se mjere performanse testnih računala u ovisnosti o njihovim vremenima obrade. Na kraju rada daje se finalni zaključak te nude se savjeti za buduću optimizaciju procesa obrade te načina spremanja podataka u okviru aplikacije.

Ključne riječi: profili brzina, zagušenja, mjerenje performansi

Extraction of traffic congestion information from large databases of GPS tracked vehicles

Abstract: This thesis describes one of the ways of loading and processing large quantities of speed profiles based on GPS data and benchmarking using application solution. It contains the speed profile explanations and description of development and user usage of specialized application whose function is to process and visually present the provided profiles. After that a series of testing is performed; the testing environment and benchmarking applications used are defined. Series of testing and performance measuring based on processing times is conducted on each of the test computers. The final part contains advices and guidelines for application improvement, processing optimizations data storing are also mentioned.

Keywords: speed profiles, congestions, benchmarking

SADRŽAJ

| | |
|---|----|
| 1. UVOD | 1 |
| 2. GPS PODACI | 3 |
| 2.1. GPS sustav | 4 |
| 2.2. GPS zapis | 5 |
| 2.3. Prikupljanje GPS podataka | 6 |
| 2.4. GPS podaci za potrebe aplikacije | 8 |
| 3. IZRADA APLIKATIVNOG RJEŠENJA ZA OBRADU I PRIKAZ PODATAKA | 12 |
| 3.1. Rječnik, Polje i Lista | 13 |
| 3.2. Tijek obrade podataka | 15 |
| 4. KORIŠTENJE APLIKATIVNOG RJEŠENJA | 18 |
| 4.1. Instalacija aplikacije i učitavanje podataka | 18 |
| 4.2. Profili brzina | 22 |
| 5. TESTIRANJE PERFORMANSI RJEŠENJA | 26 |
| 5.1. Programi za <i>benchmark</i> | 27 |
| 5.2. Računalne platforme korištene prilikom testiranja | 32 |
| 5.3. Testno okruženje i testni uzorci | 33 |
| 5.4. Testiranje | 34 |
| 5.5. Rezultati testiranja | 35 |
| 5.5.1. Testiranje referentnog dijela segmenata | 36 |
| 5.5.2. Testiranje kompletnog seta segmenata | 40 |
| 5.5.3. Zaključci testiranja | 41 |
| 6. PREPORUKE I SMJERNICE ZA DALJNI NAPREDAK | 44 |

| | |
|-------------------------------|----|
| 7. ZAKLJUČAK..... | 48 |
| POPIS LITERATURE..... | 49 |
| POPIS AKRONIMA I KRATICA..... | 51 |
| POPIS STRANIH IZRAZA..... | 52 |
| POPIS ILUSTRACIJA..... | 54 |
| POPIS PRILOGA..... | 56 |

1. UVOD

Prikupljanje GPS zapisa, bilo na globalnoj ljestvici ili lokalno, zahtjeva njihovu obradu u svrhe razumijevanja veze između predmeta praćenja i drugih traženih vrijednosti. Također, naknadna obrada podatka zahtjeva dobro poznavanje struktura zapisa te iziskuje znanja iz područja statistike kako bi se navedeni podaci učinkovito pretvorili u izvor informacija za daljnju upotrebu. Ovaj rad predstavlja primjer učinkovite obrade i korištenja sirovih profila brzina koji su zasnovani na prikupljenim GPS podacima.

Svrha diplomskog rada je obraditi i prikazati veliku količinu složenih podataka za vozila grupiranih po pojedinim segmentima prometnica unutar grada Zagreba upotrebom aplikativnog rješenja. Cilj diplomskog rada jest saznati granicu koja odgovara prihvatljivom vremenu obrade podataka programskog rješenja, definirati odnos performansi računalne snage i količine unesenih podataka u programsko rješenje te usporediti rezultate sa sličnim dosadašnjim rješenjima.

Materija diplomskog rada je izložena u sedam poglavlja:

1. Uvod,
2. Općenito o GPS podacima,
3. Izrada programskog rješenja za obradu i prikaz podataka,
4. Korištenje programskog rješenja,
5. Testiranje performansi rješenja,
6. Preporuke i smjernice za daljnji napredak te
7. Zaključak

Uvodno poglavlje opisuje strukturu rada te daje svrhu i cilj rada.

Drugo poglavlje pod nazivom *Općenito o GPS podacima* govori o tematici koja je osnova za razumijevanja daljnjih poglavlja. Definišu se pojmovi poput GPS zapisa, sustava i svrhe njihovih prikupljanja te značajnost istih.

U trećem poglavlju nazvanome *Izrada programskog rješenja za obradu i prikaz podataka* objašnjava se proces izrade aplikativnog rješenja koje je dizajnirano za obradu

specifičnih profila brzina zapisa te hijerarhijske strukture podataka. Također je obrazložen atipičan način zapisivanja vrijednosti u radnu memoriju te su navedene prednosti i mane takvog postupka.

Četvrto poglavlje, *Korištenje programskog rješenja*, vodi kroz korake korištenja aplikacije te ističe sve funkcionalnosti aplikacije i vrste primjene obrađenih podataka i profila brzina.

U petom poglavlju nazvanome *Testiranje performansi rješenja* obavlja se skup mjerenja vremena obrade preko različitih računala. Također se pokušava naći veza između performansi pojedinih komponenata računalnih sustava i brzine obrade GPS podataka.

Šesto poglavlje naziva *Preporuke i smjernice za daljnji napredak* sadržava diskusiju o provedenome testiranju te služi kao vrhunac rasprave u područjima učinkovitosti aplikacije te mogućih poboljšanja obrade profila brzina i utjecaj budućih trendova na uvjete u računalnoj snazi.

Na kraju rada nalazi se zaključak koji sadrži sažetak izvršenih testiranja te nudi razmišljanja za daljnje mogućnosti i istraživanja tematike.

Programski alat koji se koristi u diplomskom radu je *Visual Studio Community 2015* [1], a služi za izradu aplikativnog rješenja obrade podataka. Arhivski profili brzina i GPS podaci ustupljeni su od strane Fakulteta te tvrtke *Mireo d.d.* Prilozi u završnom radu su: slike, tablice, grafikoni i jednadžbe. Izvori slika, tablica i formula i jednadžbi nalaze se u *Popisu literature*. Izvori korišteni pri izradi završnog rada obuhvaćaju knjige, znanstvene članke, znanstvene radove objavljene na Internetu.

2. GPS PODACI

GPS ili Globalni Pozicijski Sustav (engl. *Global Positioning System, GPS*), kao i svi sustavi jednake razine kompleksnosti, u svakom trenutku sadrži veoma veliku količinu podataka koji se čitaju, zapisuju, analiziraju, upotrebljavaju te koriste unutar i izvan sustava. Navedeni podaci također posjeduju važnu ulogu prilikom njihove naknadne analize. Važnost potječe iz činjenice da tijekom aktivnog rada sa podacima, nije moguće u stvarnom vremenu maksimalno iskoristiti sve potencijalne informacije koje mogu sadržavati. Zbog tog razloga podaci velikog volumena se spremaju i arhiviraju upravo zbog naknadne obrade u svrhu dobivanja saznanja koja nisu očita u prvom trenutku. Takva saznanja u pravilu sadržavaju statističke podatke koji mogu mijenjati i utjecati na korištenja sustava, profila korisnika i uređaja, mjerenja napretka, personalizacije usluga i druge. Također navedene informacije znatno utječu na donošenje odluka velikog značaja i mijenjanja poslovnih strategija između mnoštvom drugih. Prednost sakupljanja podataka tijekom dužeg vremenskog perioda otkriva trendove i promjene koje se inače ne bi mogle zamijetiti zbog svakodnevnih fluktuacija te subjektivnosti osobe koje prikuplja pojedinačne dijelove.

Prostorni podaci su kao što ime govori, podaci koji sadrže prostorne ili geografske informacije i vrijednosti. Služe primarno za povezivanje vremenske i prostorne koordinate te se u te svrhe najčešće upotrebljavaju kao osnova za sustave snalaženja u prostoru, praćenja i zapisivanja puteva i ruta. U ovom radu će se govoriti o prostornim podacima kao cjelini koja sadrži neistražene informacije te će se one pokušati iskoristiti u svrhe dobivanja uvida u prometna zagušenja. Vrsta podataka o kojoj će se govoriti su profili brzina zasnovani na GPS zapisima arhivske prirode koji su sakupljeni i spremni tijekom perioda nekoliko godina. Arhivski profili brzina mogu također služiti kao statistika za buduće izračune i sustave koji djeluju u sličnim područjima i situacijama.

Sljedeća potpoglavlja trebaju dati temelj općoj terminologiji u područjima GPS sistema i zapisa te objasniti vrstu podataka koja će se koristiti u idućim poglavljima.

2.1. GPS sustav

GPS sustav je sustav satelita u orbiti zemlje koji pruža lokacijske, navigacijske i vremenske usluge. GPS je jedan od tri postojeća navigacijska i lokacijska sustava, od kojih druga dva su ruski GLONASS (rus. *Globalnaya navigatsionnaya sputnikovaya sistema, Globalni navigacijski satelitski sustav*) te europski GALILEO. GALILEO, koji trenutno sadrži 14 lansiranih satelita od planiranih 30, još uvijek nije aktivan te se osnovne mogućnosti predviđaju od 2017. godine, a puna funkcionalnost do 2020. godine. [2]

GPS sustav razvio se 1973. godine od strane Ministarstva obrane SAD-a kao primarno vojni projekt te se kao takav koristio niz godina sve do 1983. godine, gdje je postao dostupan u civilne svrhe sa smanjenom preciznošću. Tek od godine 2. svibnja 2000. godine sustav postaje dostupan u civilne upotrebe s potpunom preciznošću. Od tada pa sve do danas sustav se nadograđuje lansiranjem novijih satelita s preciznijim mogućnostima i većim kapacitetima koji mijenjaju svoje starije prethodnike.

Navedeni sustav sastoji se od 32 satelita koji se nalaze unutar 6 orbita na visini od 20180 kilometara te su njihove orbite dizajnirane na način da se u bilo kojem trenutku nalaze minimalno 3 satelita unutar horizonta korištenja bilo gdje na Zemljinoj površini. Smanjenje preciznosti GPS lociranja se može dogoditi ukoliko se prijamnik nalazi na lokaciji koja ima ograničen pogled na nebo (engl. *line of sight*), kao što je primjer u gradskim područjima sa visokim zgradama te u tunelima. [3]

U današnje vrijeme GPS sustav ima veoma široku primjenu u civilnom društvu, vojnim potrebama te područjima prometa, aeronautike, geodezije, rudarstva, građevinarstvu te ostalim sektorima i aktivnostima gdje se javlja potreba za lociranjem, snimanjem i praćenjem mobilnih i stacionarnih objekata.

2.2. GPS zapis

GPS zapis je vrsta prostornog podatka koji prati smjernice i format standardnog GPS sustava, trenutno primarnog satelitskog sustava u Ujedinjenim Američkim Državama i u Europi. Zapis također prati referentni standard WGS 84, koji stvara poveznicu između Zemljinog geoidnog oblika te, između ostalih GPS sustava za praćenje uz uvjet da je GPS sustav savršeno precizan, posjeduje točnost projekcije unutar jednog metra.

Osim što sadrži geografsku širinu i dužinu, moderni GPS zapis sadrži i pregršt ostalih podataka koji su definirani prema standardu NMEA 0183 (engl. *National Marine Electronics Association*). NMEA je dugogodišnji standard koji je uveden zbog usklađivanja GPS prijamnika, sonar uređaja, žirokompasa, autopilotnih sustava te ostalih uređaja koji komuniciraju putem prostornih GPS podataka. GPS NMEA zapis može sadržavati do 17 individualnih informacija odvojenih zarezom. Dan je primjer formata GPS zapisa koji se konformira prema NMEA standardu: [4]

\$GPXXX,V,DDDD,1,ŠŠŠŠ,2,3,4,5,6,X,7,Y, 8,9999,*Z

gdje:

- **\$** označava početak pojedinog zapisa,
- **GPXXX** označava vrstu GPS zapisa (GPSGA – detaljan satelitski zapis praćenja, GPSGV – detaljan satelitski zapis koji sadrži azimut i visinu svakog GPS satelita koji se prati, GPGST – visoka procijenjena horizontalna i vertikalna preciznost zapisa, i dr.)
- **V** označava vremenski žig u UTC formatu; sati, minute i sekunde,
- **DDDD** označava geografsku duljinu,
- **1** označava vertikalne strane svijeta (sjever ili jug)
- **ŠŠŠŠ** označava geografsku širinu,
- **2** označava horizontalne strane svijeta (istok ili zapad)
- **3** označava indikator kvalitete (1 - neispravljene koordinate, 2 - diferencijalno neispravljene koordinate, 4 – neispravljene koordinate koristeći RTK (engl. *Real Time Kinematics*) sustav, preciznost nekoliko centimetara)

- **4** označava broj satelita koji se koristio prilikom izračuna koordinata,
- **5** označava HDOP veličinu (engl. *Horizontal Dilution Of Precision*),
- **6** označava visinu antene primopredajnika na zemljinoj površini,
- **X** označava jedinicu visine (metri ili stope)
- **7** označava geoidnu separaciju antene (koristi se prilikom pretvorbe geoidnog oblika Zemlje u elipsoidni oblik kojim se služi WGS84 sustav)
- **Y** označava jedinicu prilikom izračunavanja geoidne separacije (metri ili stope)
- **8** označava starost zapisa (vrijeme proteklo nakon prijašnjeg slanja)
- **9999** označava korekcijski broj iz zemaljske stanice, ne mora biti uključen
- ***Z** označava kontrolni zbroj (engl. *checksum*)

Realni primjer GPS NMEA zapisa je sljedeći:

\$GPGGA,181908.00,3404.7041778,N,07044.3966270,W,4,13,1.00,495.144,M,29.200,M,0.10,0000*40.

Vrste NMEA poruka mogu biti različite ovisno o parametrima koje prate, kao što je vidljivo iz *GPG* segmenta. *\$GPGGA* poruke su među najzastupljenijima te u većini slučajeva, za precizno lociranje i dobivanje GPS lokacija i informacija potrebno je više vrsta GPS zapisa, kao što je primjerice *\$GPGGA* i *\$GPGSV*. U slučajevima preciznoga lociranja i kartiranja, *\$GPGST* zapisi su posebice korisni jer sadrže koordinate, korekcijske podatke te horizontalne i vertikalne standardne vrijednosti devijacije, parametri koji se intenzivno koriste pri praćenju zemaljskih predmeta. [4]

2.3. Prikupljanje GPS podataka

Postoji mnogo potreba i prednosti prikupljanjem velike količine podataka iste vrste te je to slučaj i s GPS zapisima. Zavisno o vrsti te predmetu praćenja, sakupljanje GPS podataka se obavlja sa svrhom statističkih saznanja, informacija o kretanju te uvida u optimizaciju ruta, između ostaloga. Statističke osnove i setovi podataka koji proizlaze iz zapisivanja velikog broj GPS podataka čine značajan izvor informacija za dizajniranje

programa, uređaja i sustava koji ih koriste te pogotovo za sustave predikcije budućih prostornih događaja koji svoje zaključke deriviraju iz spomenutih podataka. Između mnogih primjena zapisivanja i korištenja velikih količina GPS podataka, navedeno je nekoliko primjera koji najbolje odražavaju značajnost rada sa spomenutim setovima podataka.

Primjerice, *Urbano* i *Cagnaci* [5] koriste praćenje GPS podataka za potrebe konstruiranja baze prostornih podataka koja prati kretanje životinja. Također predlažu rješenja i odgovore na pitanje kako najučinkovitije iskoristiti sirove prostorne podatke s obzirom na životinjske vrste, staništa te ostale parametre sadržane unutar GPS zapisa.

Također, *Czerniak* i *Genrich* [6] koriste GPS podatke pri integraciji sa GIS sustavima (engl. *Geographic Information System*) za kartiranje i optimizaciju geoinformacijskih sustava te nadogradnju navedenih s prostornim značajkama deriviranima iz GPS zapisa.

Markham [7] je napravila istraživanje koja objašnjava učinkovitost programa koji prati uzorke kretanja vozača osobnih vozila s dijagnosticiranim poremećajem hiperaktivnosti i deficita pažnje (engl. *Attention deficit hyperactivity disorder, ADHD*). Dijagnosticirane osobe su pristale na praćenje njihovih dnevnih navika vožnje i problema u prometu s kojima se susreću tijekom određenog perioda vremena. Setovi GPS podataka koji su se prikupljali za vrijeme testnog programa vozača pomogli su pri otkrivanju uzorka ponašanja te daljnjeg istraživanja osoba s dijagnosticiranim poremećajem. Također, uspješni postupci liječenja ADHD-a su razvijeni zahvaljujući spomenutim podacima.

Sukladno ostalim primjerima, tvrtka Mireo d.d. prikuplja i koristi GPS podatke u svrhu dobivanja profila brzina koje zatim koristi za poboljšanja navigacijskih sposobnosti svoje GPS aplikacije *Mireo DON'T PANIC* [8] koja, osim klasične navigacije, upotrebljava veoma brzo pretraživanje, odabir i preračunavanje ruta. U partnerstvu sa Fakultetom prometnih znanosti 2016. godine razvili su adaptivni model Dijkstra algoritma koji koristi navedene podatke za dinamičnu promjenu rute pri navigaciji, o čemu će biti više rečeno u idućem potpoglavlju.

2.4. GPS podaci za potrebe aplikacije

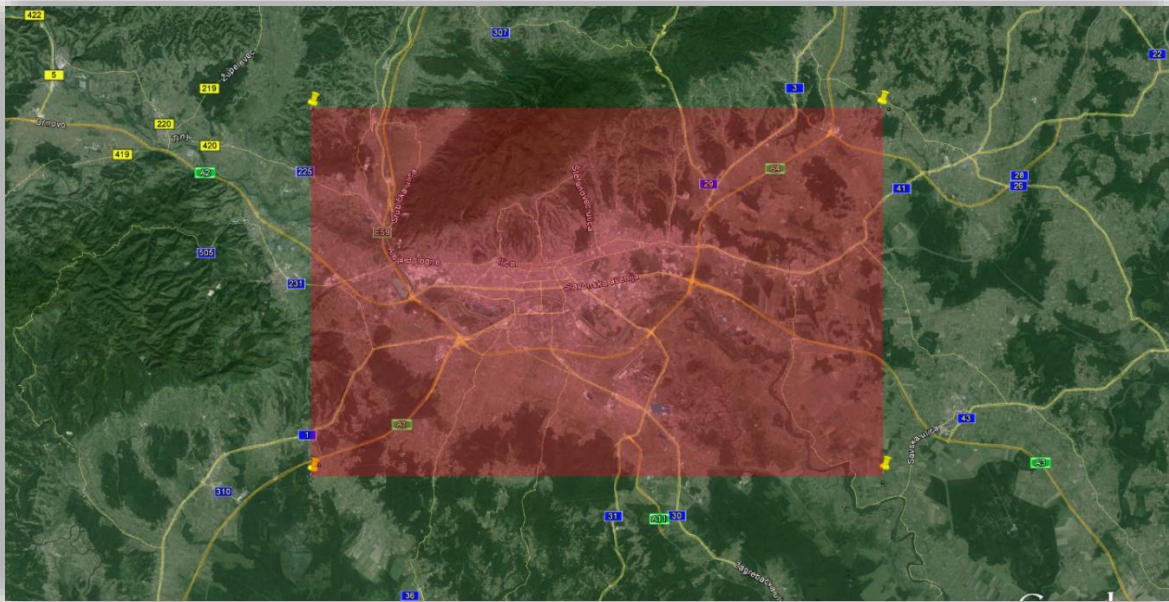
Za potrebe aplikativnog rješenja ovog rada koristit će se već dostupni profili brzina koji su zasnovani na GPS podacima praćenja vozila u Hrvatskoj te posjeduju određenu razinu inicijalne obrade te su strukturirani i standardizirani na način koji omogućava izravnije bavljenje njihovom obradom i kasnijim korištenjem.

Projekt SORDITO (*Sustav za Optimizaciju Ruta u Dinamičkom Transportnom Okruženju*) zajednički je projekt između Fakulteta prometnih znanosti i tvrtke *Mireo d.d.* u trajanju od listopada 2014. godine do početka 2016. godine. Cilj projekta je bio izraditi sustav optimizacije ruta vozila koji mijenja rutu dinamički ovisno o prometnim opterećenjima u prometu. Da bi takav izračun bio moguć, potreban je dovoljan uzorak i rezolucija podataka koji bi služili kao osnova za predikciju budućih trendova brzina. Partnerska tvrtka *Mireo d.d.* uz ostale aktivnosti, prikupljala je GPS tragove približno 4200 vozila u razdoblju od 5 godina te je u navedenom razdoblju akumulirala oko 7 milijardi zapisa [8]. Navedeni podaci su predstavljali temelj zajedničkom projektu te derivatni oblik jednog njihovog dijela predstavlja temelj za izradu aplikacije u ovom radu. Za potrebe aplikativnog rješenja, odabrano je referentno područje grada Zagreba.

Kao što je priloženo na slici 1, područje zahvaća kompletnu cjelinu grada te iznosi otprilike 1350 četvornih kilometara. Unutar navedenog područja nalazi se 47.689 cestovnih segmenata. Jedan segment obuhvaća određeni dio prometnice duljine od nekoliko desetaka metara pa do reda veličine 100 metara, odnosno dio prometnice između dva susjedna križanja. Također, svaki segment posjeduje svoj jedinstveni identifikacijski broj, geografske koordinate početne i završne točke te prometni profil brzina tokom svakog dana u tjednu.

Nadalje, segmenti su za svrhe rada prerađeni u oblik prihvatljiviji aplikativnoj obradi. Svaki segment sadrži podatke za sedam dana u prosječnom tjednu te ukoliko se prometnicom zahvaćenom određenim segmentom prometuje u oba smjera, podaci sadrže odvojene profile brzina za svaki smjer.

Zbog malog broja zabilježenih vozila tijekom noćnih sati mjerenje profila brzina je sadržano u dva intervala: jutarnji interval od 5 sati i 30 minuta pa sve do 22 sata te noćni interval od 22 sata pa do 5 sati i 30 minuta. Za noćni interval profila brzina je uzeta prosječna vrijednost svih izmjerenih brzina te je nazvan *free flow interval* odnosno interval slobodnog toka jer se podrazumijeva normalan tok prometa u navedenom razdoblju.



Slika 1. Odabrano područje grada Zagreba kao izvor profila brzina

Hijerarhijska struktura podataka je dana na sljedeći način. Svaki cestovni segment posjeduje svoju mapu koja je nazvana kao identifikacijski broj (ID) segmenta. Svaka takva mapa sadrži sedam ili četrnaest datoteka koje odgovaraju sedam dana unutar prosječnog tjedna; četrnaest datoteka je sadržano u slučaju dvosmjerne prometnice na području segmenta. Uz navedene datoteke nalazi se i jedna datoteka koja sadržava opće informacije o segmentu te vrijednost noćnog intervala.

Segmenti koji će se koristiti u radu za potrebe obrade sadrže nekoliko podataka koji će se koristiti. Tablica 1. prikazuje primjer *.txt* datoteke koja odgovara vrijednostima prosječnih brzina tokom jednog dana te predstavljaju jedan dan određenog segmenta.

Prvi stupac nazvan *TIME* sadrži vremenski zapis izražen u sekundama s intervalom od 300 sekundi (5 minuta) te će se navedene vrijednosti koristiti prilikom obrade podataka.

Tablica 1. Primjer datoteke sa profilima brzina

| TIME | SMOOTH | REL | ABS | DATA_PER_INT |
|-------|--------|-------|-------|--------------|
| 19800 | 101.9 | 107.7 | 101.2 | 11.0 |
| 20100 | 102.0 | 107.7 | 101.2 | 11.0 |
| 20400 | 102.2 | 107.7 | 101.2 | 11.0 |
| 20700 | 102.4 | 101.0 | 94.8 | 13.0 |
| 21000 | 102.6 | 101.0 | 94.8 | 13.0 |
| 21300 | 102.7 | 101.0 | 94.8 | 13.0 |
| 21600 | 102.9 | 105.4 | 99.0 | 7.0 |
| 21900 | 103.1 | 105.4 | 99.0 | 7.0 |
| 22200 | 103.3 | 105.4 | 99.0 | 7.0 |
| 22500 | 103.5 | 98.5 | 92.5 | 15.0 |
| 22800 | 103.8 | 98.5 | 92.5 | 15.0 |
| 23100 | 104.0 | 98.5 | 92.5 | 15.0 |
| 23400 | 104.2 | 118.3 | 111.1 | 16.0 |
| 23700 | 104.5 | 118.3 | 111.1 | 16.0 |
| 24000 | 104.8 | 118.3 | 111.1 | 16.0 |
| 24300 | 105.1 | 96.5 | 90.6 | 25.0 |
| 24600 | 105.4 | 96.5 | 90.6 | 25.0 |

Sljedeća tri stupca sadržavaju profile brzina koji su prilagođeni različitim prikazima i potrebama SORDITO projekta te su izraženi u kilometrima na sat (km/h). Drugi stupac imena *SMOOTH* sadržava profil brzine koji je relativno zaglađen pomoću metode *cubic smoothing spline* u svrhe lakšeg grupiranja profila brzina. Navedena metoda se koristi u slučajevima grafičkog zakrivljivanja vrijednosti [9] ako odabrane vrijednosti izražavaju kaotičan karakter. Treći stupac imena *REL* sadržava vrijednosti omjera noćne brzine te apsolutne koje su računane formulom:

$$\frac{\text{"free flow"}}{ABS} * 100 \quad [1]$$

Četvrti stupac nazvan *ABS* predstavlja apsolutne izmjerene vrijednosti te će se koristiti prilikom izračunavanja profila brzina. Peti stupac nazvan *DATA_PER_INT* pokazuje broj individualnih mjerenja unutar petominutnog intervala te je vrijednost stvarno izmjerene brzine jednaka prosječnoj vrijednosti mjerenja navedenom u navedenom stupcu.

Također postoji veliki broj „praznih mjerenja“; segmenata čiji GPS zapisi su dolazili u dovoljno dugačkom intervalu (par sati) da je cijeli profil brzine jedna velika prosječna vrijednost individualnih mjerenja tokom odabranog dana. Takve datoteke će prikazivati jednoličnu brzinu tijekom cijelog dana.

Navedene vrijednosti će se koristiti kao temelj izračuna i vizualizacije prilikom izrade i korištenja aplikativnog rješenja, o čemu će biti riječi u idućem poglavlju.

3. IZRADA APLIKATIVNOG RJEŠENJA ZA OBRADU I PRIKAZ PODATAKA

Za potrebe ovog rada, dizajnirana je aplikacija *SegCalc* koja obrađuje podatke iz navedenih sirovih oblika u one kojima se može lako manipulirati te iz kojih se jasno mogu izvući zaključci prilikom promatranja prometnih opterećenja.

Aplikacija *SegCalc* je zasnovana na C# programskom jeziku te Windows Formama (engl. *Windows Forms*, *WinForms*), nudi lak način za izradu grafički orijentiranog sučelja (engl. *GUI*, *Graphical user interface*). Windows Forme posjeduju sve što je potrebno za aplikacije čija svrha je obrada jednostavnih i kompleksnih setova podataka te su opremljene s vizualnim alatima koji omogućavaju relativno jednostavan pregled zadanih podataka.

Integrirano razvojno okruženje *Microsoft Visual Studio Community* verzija 2015; inačica 14.0.25123.00, je korišteno za potrebe razvoja aplikativnog rješenja. VSC je općepriznato moćno okruženje koje nudi pregršt mogućnosti unutar C# kao i lepeze ostalih programskih jezika. Koristi se za razvijanje računalnih programa za Windows operacijski sustav, web stranice, aplikacije i usluge.

Prije osvrta na unutarnje mehanike aplikacije i detaljne načine obrada podataka, objasniti će se metodike razmišljanja prilikom planiranja izrade aplikacije i primarno načina zapisivanja podataka. Za razliku od „klasičnih“ rješenja koje za potrebe strukturiranih vrijednosti koriste baze podataka, aplikacija *SegCalc* će zapisivati svoje vrijednosti unutar hijerarhijskih struktura koje se nalaze u radnoj memoriji računala. Prednosti i mane takvog načina zapisivanja su sljedeće.

Prvi i osnovni razlog je lokacija spremanja podataka prilikom korištenja programa. Baze su organizirane zbirke podataka koji se primarno nalaze na mjestu trajnog skladištenja; tvrdom disku. Iako baze podataka posjeduju veoma visoku razinu hijerarhije i uređenosti podataka te pogotovo su subjekti naprednim i moćnim pretraživanjima pomoću SQL (engl. *structured query language*) naredbi, u suštini njihova brzina obrade ovisi o performansama tvrdog diska. Doduše, dobra strana takvog načina pohranjivanja

je da je trajan; pohranjivanjem na tvrdi disk podaci ostaju spremljeni nakon gašenja računala i komponenti. Dodatna olakotna okolnost prilikom korištenja baze podataka u trenutnom kontekstu je sve šira pojava takozvanog *Solid State* diska (engl. *Solid State Drive, SSD*). Performanse zapisivanja i čitanja podataka na takvim vrstama tvrdih diskova su nekoliko faktora bolje od konvencionalnih tvrdih diskova.

No, u trenutku pisanja rada klasični tvrdi diskovi su još uvijek zastupljeni u većoj mjeri, pogotovo u uredskim okruženjima, te je radna memorije odabrana kao mjesto zapisivanja privremenih podataka. Aplikacija je rađena sa time na umu, te u trenutku pisanja rada, performanse obrade podataka su znatno bolje korištenjem tipova podataka koji se pohranjuju unutar radne memorije. O performansama ovakvog načina spremanja podataka biti će riječi u četvrtom poglavlju ovoga rada.

3.1. Rječnik, Polje i Lista

Nakon što je definirano područje spremanja varijabli tijekom njihove obrade i korištenja, potrebno je odabrati načine spremanja te dohvaćanja setova i skupova podataka koji moraju biti logički grupirani i povezani. U tu svrhu koristi se nekoliko vrsta „spremnika“ koji moraju biti u mogućnosti sadržavati velike količine podataka te u isto vrijeme omogućiti pretraživanje što učinkovitijim i vremenski osjetljivijim.

Za navedene potrebe se koristi Rječnik (engl. *Dictionary*) i Lista (engl. *List*) kao glavni elementi sadržavanja podataka. Rječnik je sustav uređenih parova ključeva i vrijednosti gdje je moguće povezati određene vrijednosti sa određenom ključnom vrijednosti ili izrazom. Primjer rječnika bi bilo povezati broj „5“ sa pojmom „jabuka“. Kada bi pretraživali Rječnik za broj 5, dobili bi vrijednost „jabuka“. Navedeni mehanizam djeluje veoma jednostavno, no svoju pravu moć pokazuje kada sadržava tisuće ključeva gdje su vrijednosti indeksirane po svojim ključevima te je pretraživanje istih koristeći ključne vrijednosti veoma brzo. [10]

Sljedeći element pohranjivanja velikog podataka je Lista. Prilikom klasičnog spremanja nekoliko vrijednosti koje su međusobno vezane u pravilu se koriste varijable

Polja (engl. *Array*). Polja su prostorne varijable koje sadrže više vrijednost unutar sebe koje su označene indeksom. Jedina prednost kod Polja je brže vrijeme dohvaćanja podataka unutar pojedinačnih indeksa. Nedostatci polja su unaprijed definirane veličine polja; ukoliko se pokušava spremati vrijednost u indeks koji prelazi broj indeksa polja, javlja se pogreška (engl. *IndexOutOfRangeException*) [11] prilikom rada, te sporost u radu sa višedimenzionalnim načinima spremanja vrijednosti. Također, Polja mogu sadržavati samo jednu vrstu podatka. Ako se Polje primjerice deklarira za pohranu brojeva, više nije moguće unutar njih dodavati znakove ili slova. Zbog tih razloga se koristi Lista, koja pak sadrži nekoliko prednosti:

- Mogućnost dodavanja više vrsta podataka (*int, double, string, datetime...*)
- Višedimenzionalnost Liste, veoma jednostavan kod za unos
- Dinamička veličina Liste; nije potrebno unaprijed definirati listu, dodavanjem vrijednosti njen indeks se automatski povećava.
- Vežanjem Klasa za Listu povećavaju se mogućnosti upotrebe i fleksibilnosti Listi
- Veoma laka upotreba i dodavanje vrijednosti, pogotovo prilikom višedimenzionalnih Listi

Polja će se koristiti u nekoliko iteracija kroz pisanje programa, no na dijelovima gdje se ističu njihove prednosti te gdje nema druge alternative.

Korištenje određenih vrsta tipova varijabli je namjerno odabrano u svrhu boljih performansi prilikom manipulacije podataka, tako da postoji nekoliko izbora koje valja spomenuti. Kombinacija Rječnika i Listi mnogo brža pri obradi od korištenja baze podataka zbog nekoliko razloga.

Korištenjem Rječnika i Liste, podaci spremeni u navedene okvire su cijelo vrijeme pohranjeni u radnu memoriju (engl. *RAM, Random Access Memory*) koja posjeduje veoma visoke brzine prijenosa podataka u odnosu na tvrdi disk. Negativna strana navedenog načina spremanja podataka jest da nakon što se računalo ugasi, svi podaci se gube. No, činjenica da je aplikativno rješenje zamišljeno na principima mobilnog karaktera i privremenog korištenja na pojedinom računalu čini navedenu negativnu stranu takvog načina zapisivanja podataka nebitnom.

3.2. Tijek obrade podataka

Unutarnji mehanizmi aplikativnog rješenja funkcioniraju na sljedeći način. Nakon odabira podataka za obradu, program počinje sa učitavanjem. Obavlja se skeniranje svih podmapa i datoteka unutar odabrane mape, te se adrese svake datoteke spremaju u Polje znakovnih vrijednosti (engl. *string array*). Zatim se koristi glavna petlja koja prolazi kroz svaki element polja, odnosno koja učitava svaku datoteku na navedenoj adresi.

Prilikom učitavanja pojedine datoteke, ime mape, odnosno ime segmenta ceste se sprema u privremenu varijablu koja će se pridružiti elementima Liste. Nakon što je datoteka učitana, petlja prolazi kroz svaki njen redak. Podaci unutar datoteke su odvojeni tabulator znakom (engl. *tab*) te aplikacija pomoću pretraživanja navedenog znaka odvaja pojedinačne vrijednosti u retku i sprema ih u privremeno polje. Elementi tog polja, zajedno sa imenom segmenta, se spremaju u Listu. Postupak je ilustriran u tablici 2.

Svaka datoteka sadrži 198 linija te svaka linija predstavlja jedno mjerenje u inkrementu od 300 sekundi, odnosno 5 minuta. Tako da se prvo mjerenje događa u 19800 sekundi, ili 5 sati i 30 minuta ujutro, a zadnje mjerenje u 78900 sekundi, odnosno 22 sata. Za potrebe pojednostavljenja dinamičkog grafikona, te vrijednosti su sažete u inkremente od 15 minuta te odgovarajuća Lista za svaki dan sadrži 66 elemenata.

Nakon što petlja procesira sve linije jedne datoteke, Listu sprema u Rječnik druge razine, gdje ključna vrijednost Rječnika iznosi ime datoteke (npr. „194222_Day_0_Cat_2“) a njegova vrijednost je Lista. Broj rječnika je dakle jednak broju datoteka unutar jedne mape a taj broj može iznositi 7 (svaki dan u tjednu), ako segment sadržava podatke za samo jedan smjer, ili 14 u slučaju podataka za oba smjera.

Tablica 2. Proces pretvorbe „sirovih“ podataka u Listu

| Segment = 194222 | | | | |
|------------------|--------|-------|-------|--------------|
| TIME | SMOOTH | REL | ABS | DATA_PER_INT |
| 19800 | 101.9 | 107.7 | 101.2 | 11.0 |
| 20100 | 102.0 | 107.7 | 101.2 | 11.0 |
| 20400 | 102.2 | 107.7 | 101.2 | 11.0 |
| 20700 | 102.4 | 101.0 | 94.8 | 13.0 |



| | aray[0] | aray[1] | aray[2] | aray[3] | aray[4] |
|---------|---------|---------|---------|---------|---------|
| line[0] | 19800 | 101.9 | 107.7 | 101.2 | 11.0 |
| line[1] | 20100 | 102.0 | 107.7 | 101.2 | 11.0 |
| line[2] | 20400 | 102.2 | 107.7 | 101.2 | 11.0 |
| line[3] | 20700 | 102.4 | 101.0 | 94.8 | 13.0 |



| List [0] <Entry> | |
|------------------|-----------|
| Segment | 194222 |
| Time | array [0] |
| Speed | array [2] |



| List [1] <Entry> | |
|------------------|-----------|
| Segment | 194222 |
| Time | array [0] |
| Speed | array [2] |



| List [2] <Entry> | |
|------------------|-----------|
| Segment | 194222 |
| Time | array [0] |
| Speed | array [2] |

Kada se obrada svih datoteka unutar jedne mape završi, svi Rječnici se stavljaju u Rječnik prve razine te je njegova ključna vrijednost jednaka imenu segmenta (primjerice „194222“), a vrijednosti vezane za taj ključ su sedam ili četrnaest rječnika, iznos koji korespondira svakom danu u tjednu te uvjetu je li promet zabilježen u jednom ili dva smjera. Hijerarhijski dijagram opisane strukture se može detaljnije vidjeti na tablici 3.

Tablica 3. Hijerarhijska struktura podataka

| Rječnik prve razine | | | | |
|----------------------|----------------------------|------------------------|----------------|------------|
| Ključ | Vrijednost | | | |
| 194225 --> | Riječnik druge razine | | | |
| 194226 | Ključ | Vrijednost | | |
| 194227 | 194225_Day_0 --> | Lista | | |
| 194228 | 194225_Day_1 | Indeks Liste | Elementi Liste | |
| 194229 | 194225_Day_2 | Lista[0] --> | Segment | "194228" |
| 194230 | 194225_Day_3 | Lista[1] | Vrijeme | 19800 |
| 194231 | 194225_Day_4 | Lista[2] | Brzina | 116.9 Km/h |
| ... | ... | ... | | |

Obrada podataka asinkrono odvojena od funkcionalnosti sučelja; navedena odluka poboljšava brzinu obrade podataka jer program može obrađivati podatke u jednoj procesorskoj niti, te čekati korisnika u drugoj niti koja je zadužena za kontrole sučelja. Navedeni način omogućava veoma brzo i učinkovito dohvaćanje elemenata liste. Iako je velik dio dizajniranja aplikacije orijentiran na performanse i učinkovitost aplikacije prilikom obrade podataka, prilikom učitavanja velikog broja segmenata vrijeme obrade nije neznatno.

U sljedećem poglavlju biti će riječi o svim aspektima korištenja programskog rješenja. Također će se govoriti vizualnoj prezentaciji učitanih segmenata te će se objasniti mehanizmi prikazivanja prometnih zagušenja.

4. KORIŠTENJE APLIKATIVNOG RJEŠENJA

Korištenje aplikativnog rješenja *SegCalc* je zamišljeno da bude jednostavno i intuitivno, kao što je za očekivati iz svake moderne aplikacije današnjice. Aplikacija se sastoji od dva segmenta: učitavanje profila brzina te manipulacija istima. U prvom segmentu aplikacije odabiru se mape sa podacima te se učitavaju u aplikaciju, obrađuju te pohranjuju na radnu memoriju. Drugi segment aplikacije bavi se vizualnom prezentacijom učitanih i obrađenih podataka.

Prilikom dizajniranja aplikacije, trud je uložen u ostvarivanje pozitivnog korisničkog iskustva. Sučelje je minimalističko te se nastojalo grupirati sve osnovne kontrole prema zajedničkim svojstvima i funkcijama. Također je velik trud uložen u skaliranje prozora sa koordinatama učitanih segmenata sa svrhom što vidljivijih prometnica u odnosu na učitane podatke. Sljedeća poglavlja će objasniti detaljniju instalaciju te korištenje aplikativnim rješenjem.

4.1. Instalacija aplikacije i učitavanje podataka

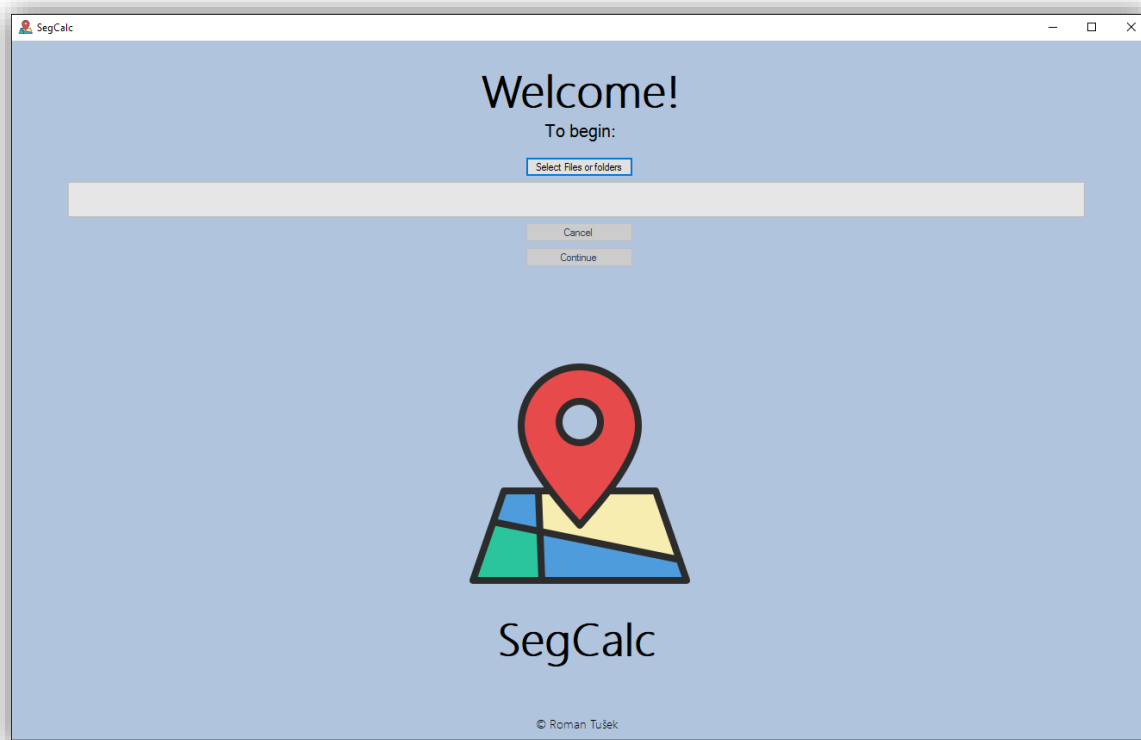
Da bi se aplikacija mogla koristiti, potrebno je obaviti instalaciju iste te provjeriti da li računalo zadovoljava sve preduvjete za njen nesmetan rad.

Instalacijsku mapu aplikacije je potrebno prebaciti s eksternog medija na lokaciju po želji unutar osobnog računala te nakon toga se pokretanje vrši duplim klikom na izvršnu datoteku unutar mape. *Visual Studio* prilikom izdavanja aplikacije nudi jednostavan, ali moćan instalacijski paket koji prije pokretanja aplikacije provjerava sustav računala za instalirane komponente operacijskog sustava. Ukoliko nedostaju, automatski se pojavljuje prozor za dohvaćanje istih sa interneta. S time na umu, aplikaciju je poželjno instalirati na osobno računalo koje posjeduje pristup internetu tijekom prvog pokretanja. U najvećem postotku slučajeva, komponente koje nedostaju su *.NET* okviri (eng. *.NET Framework*), skup programskih biblioteka koje nude podršku mnogim programskim jezicima koji se izvršavaju na računalu. U slučaju aplikativnog rješenja koje se izrađuje u ovom radu, potrebno je imati instaliranu verziju 4.5.2 *.NET* okvira. Aplikacija *SegCalc* je uspješno

instalirana i pokrenuta na svim testnim računalima. Tijekom obrade nikada nije došlo do pogreške u programu te je aplikacija prikazala zadovoljavajuću razinu stabilnosti.

Nakon uspješne instalacije, prikazuje se glavni prozor aplikacije prikazan slikom 2. U takozvanom uvodnom prozoru, korisnik učitava segmente po želji. Odabir je moguć putem dijaloga za odabir mapa. Zatvaranjem dijaloga za odabir, obrada podataka se pokreće te se korisnika obavještava o stupnju obrade putem trake za napredak (engl. *progress bar*). Tijekom obrade korisnik može prekinuti proces pritiskom na gumb za prekid (*cancel*). Navedena vremena obrade podataka će biti primarni fokus sljedećeg poglavlja, gdje će se obaviti testiranje na više računalnih platforma te prikazati ovisnosti performansi komponenata i vremena obrade.

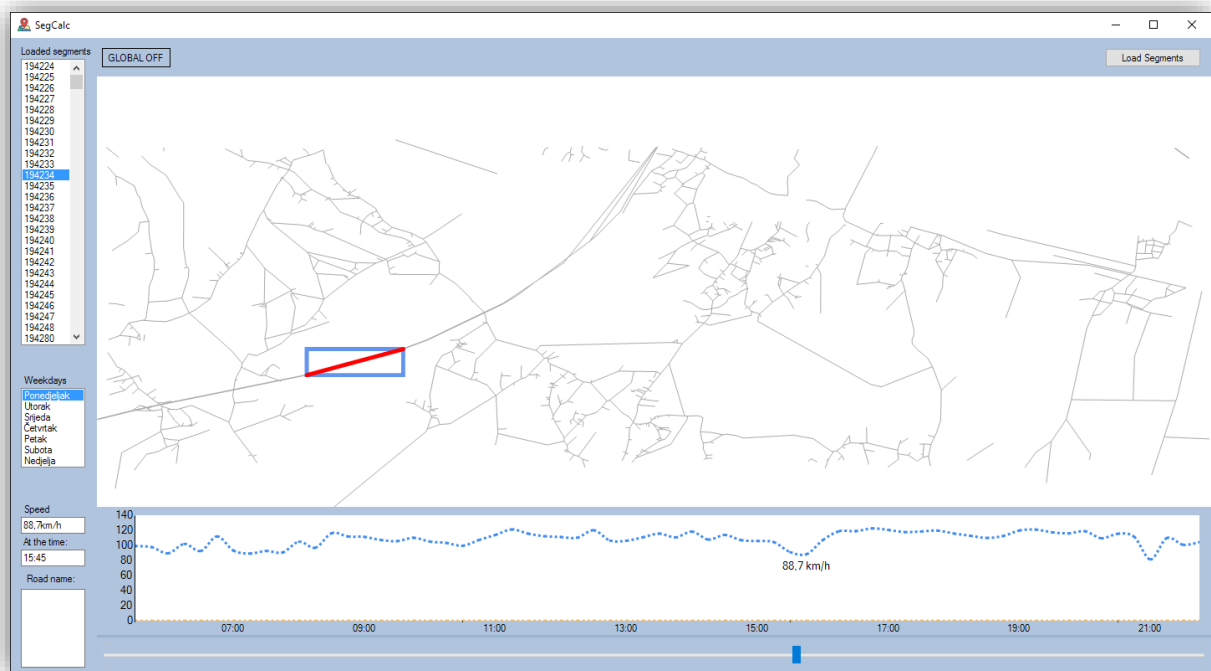
Završetkom obrade prikazuje se vrijeme obrade te se otključava gumb za nastavak na idući segment korištenja.



Slika 2. Uvodni program aplikativnog rješenja *SegCalc*

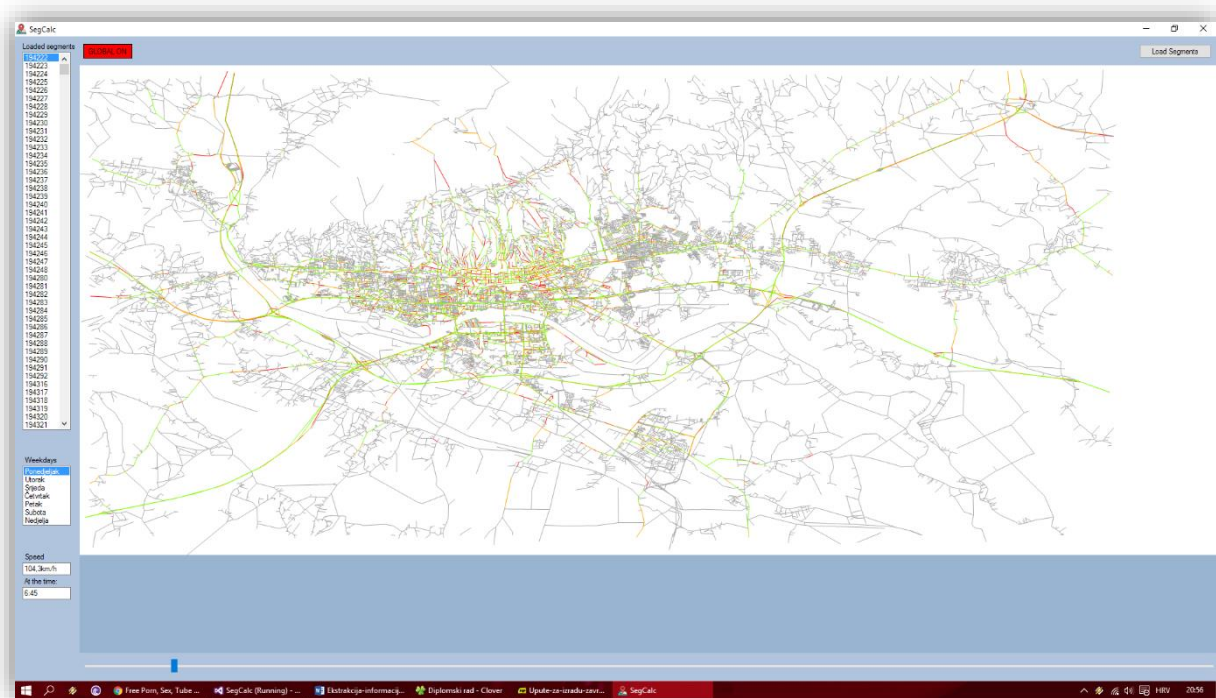
Pritiskom gumba, otvara se glavni panel. Panel se sastoji od nekoliko dijelova:

- Traka za izbor načina prikaza te gumb za učitavanje novih segmenata (gornji rub)
- Glavno platno (sredina),
- Kontrola odabira segmenata i dana u tjednu (lijevi rub),
- Kontrola odabira vremena u danu (donji rub) te
- Grafički prikaz profila brzina (iznad kontrole odabira vremena u danu). (slika 3.)



Slika 3. Glavni prozor aplikacije sa naglašenim kontrolama

Glavno platno prikazuje geografsko područje nastanjeno sa odabranim segmentima prometnica. Kontrole uz lijevi i donji rub prozora omogućavaju korisniku da odabere pojedinačni segment te dan u tjednu čije vrijednosti brzina želi vidjeti. Profili brzina zatim se vizualno prikazuju na vremenskom grafikonu, te ih je moguće pojedinačno označiti pomoću kontrole klizača (engl. *slider*; *trackbar control*). Odabrani segment će automatski biti označen na platnu čime je olakšano pretraživanje segmenata te omogućena vizualna veza između analitičkih atributa prometnice i njene geografske lokacije.



Slika 4. glavni prozor aplikativnog rješenja *SegCalc*

U gornjem lijevom rubu nalazi se globalni gumb, obojan žarko crvenom bojom zbog lakše uočljivosti i raspoznavanja perspektiva rada. Pritiskom na navedeni gumb, mijenja se kontrola odabira te umjesto odabranog jednog segmenta, korisnik može odabrati sve učitanе segmente te na taj način vidjeti prometna zagušenja na višoj geografskoj razini. Takav način pogleda je posebice pogodan i koristan za vizualno identificiranje zona zagušenja pojedinačnih ulica, križanja i čitavih gradskih blokova. Slika 4. prikazuje primjer takvog slučaja uporabe, gdje je aplikacija učitala, obradila i prikazala kompletan set odabranih prostornih podataka, veličine 3.72 gigabajta. Boja, tj. oznaka zagušenja ovisi o prosječnoj, maksimalnoj i minimalnoj brzini tokom dana te se dinamički izračunava pojedinačno za svaki segment, što će biti objašnjeno u sljedećem potpoglavlju.

4.2. Profili brzina

Bitan aspekt prilikom vizualizacije prostornih podataka unutar aplikacije jest interakcija s prometnim segmentima te razumijevanje vrijednosti i vizualnih obavijesti prilikom mijenjanja vremenskih parametara.

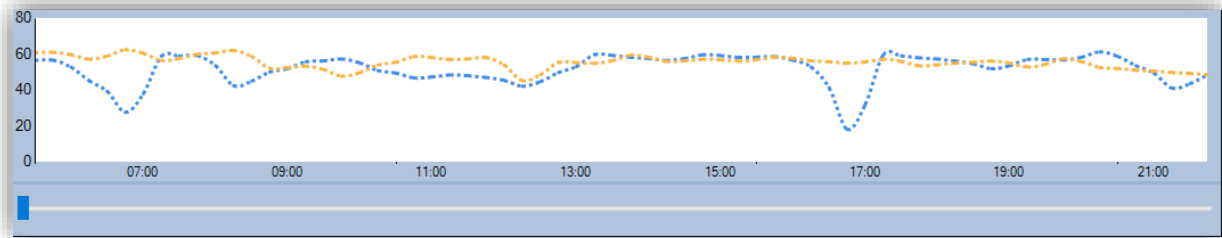
Uzeći u obzir da prilikom planiranja prometnica u pravilu se uzima veoma malo ili nimalo vršnih razdoblja u obzir, posljedica toga je očita. Smanjene brzine vozila su česta pojava, pogotovo u područjima raskrižja, gdje se pojavljuje najveći broj presijecanja prometnih linija te samim time najveća smanjenja brzina vozila koja prometuju navedenim prometnicama. Dakako, mnogo faktora utječe na stvarno zagušenje i smanjenje nominalnih brzina vozila na pojedinim raskrižjima u vršnim razdobljima:

- Kapacitet prometnice,
- Broj prometnih traka,
- Gustoća prometna,
- Lokacija u odnosu na glavne prometne pravce,
- Broj prometnica koje se križaju i dr.,

te se prilikom planiranja izgradnje novih prometnica ili prilikom obnove postojećih nastoji uzeti sve navedeno u obzir. No, entropija sustava u stvarnom vremenu te pogotovo s razinom kompleksnosti poput one koje posjeduje zagrebački prometni sustav je veoma visoka te nije uvijek osjetljiva na utjecaj koji je očekivan. Rezultat navedenih parametara uzrokuje neželjena smanjenja brzina vozila na prometnicama.

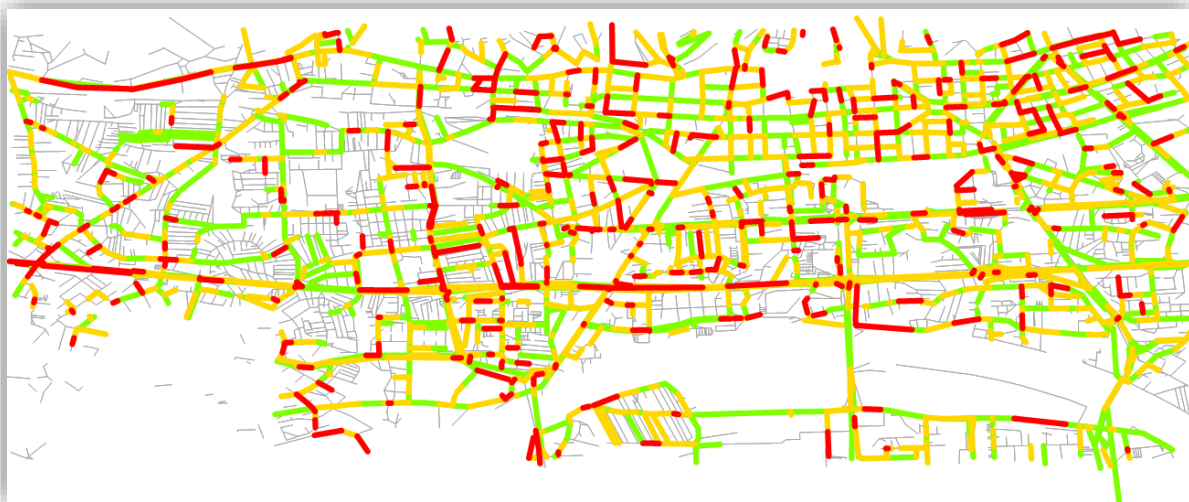
Promatrajući dnevne profile mnogih prometnica u gradu Zagrebu u pravilu se pojavljuju dvije glavne točke zagušenja, tj. vršni sati: jutarnja i popodnevna. Jutarnja točka zagušenja nastaje u periodu od 6:45 sati do 7:45 sati te predstavlja „jutarnju gužvu“, vrijeme kada većina stanovnika grada Zagreba komutira na radna mjesta. Zbog činjenice da je većina radnih poslova prijedodnevnog vremenskog karaktera (primjerice 7-15h ili 8-16h u osmosatnim smjenama) u navedenim jutarnjim periodima se pojavljuje veoma mnogo vozila na zagrebačkim prometnicama. Recipročno tome, „popodnevna gužva“

nastaje prilikom komutacije vozila sa radnog mjesta na mjesta stanovanja, te se javlja u intervalu od 16:15h do 17:15h. Veoma reprezentativni primjer takvih slučaja zagušenja prikazan je na slici 5. koja prikazuje profile brzina te smanjenja brzina vozila u navedenim vremenskim intervalima. [21]



Slika 5. Grafikon dvosmjernih profila brzina sa jasno prikazanim vremenima zagušenja unutar aplikacije *SegCalc*

Prilikom geografskog prikazivanja vizualnih oznaka brzina segmenata potrebno je pridodati navedene vrijednosti segmentima. To je učinjeno učitavanjem datoteke sa koordinatama svih ustupljenih segmenata te jednostavnim uspoređivanjem imena učitanih sa imenima odabranih segmenata. Nakon što se pojedini segment pronađe, vrijednosti koordinata njegovog početka i kraja se spremaju u Listu zajedno sa ostalim podacima segmenta. Boja segmenata ovisi o trenutnoj brzini koja je postignuta u trenutku označenom kontrolom klizača. Definicija brzine zagušenja prometnog toka je definirano na sljedeći način. Za svaki prometni segment izračunat je interval između maksimalne i minimalne brzine vozila unutar profila brzina. Time se smatra da je segment u zagušenju ako se brzina vožnje nalazi unutar donjih 10% vrijednosti intervala. Srednja brzina „polu-zagušenja“ se smatra vrijednost brzina između donjih 10 i 20% vrijednosti intervala. Brzina normalnog toka paralelno tome smatra se svaka brzina iznad donjih 20% vrijednosti intervala brzina. Boje, kao što je i objašnjeno, prikazuju prosječnu brzinu vozila u danom trenutku odnosno prikazuju prometno zagušenje u slučaju crvene boje, usporenu vrijednost brzine vožnje u slučaju žute boje te normalnu brzinu u slučaju zelene boje, kao što je prikazano slikom 6.



Slika 6. Prikaz zagušenja centra grada Zagreba

Potrebno je naglasiti kako velik broj prometnih segmenata ne sadrži diferencirajuće prometne profile tj. broj mjerenja je dovoljno rijedak da se smatraju kandidatima za uprosječnjivanje dnevnih vrijednosti. Takvi segmenti sadrže jednoličnu te linearnu vrijednost brzina tijekom cijelog dana te nisu od upotrebe u aplikaciji. Prilikom klasificiranja vrsta zagušenja, proces provjerava ukoliko segment sadrži veoma mali ili nikakav interval brzina (u slučaju navedenih vrsta segmenata) te ga označava sivom tankom bojom. Segment je prikazan da daje dodatno okruženje i snalaženje unutar topologije prikaza, no sam po sebi ne prikazuje brzinske vrijednosti jer ih niti ne posjeduje.

Navedene mehanizme je posebno zanimljivo promatrati prilikom učitavanja velikog broja segmenata. Tada je moguće primijetiti tzv. „disanje“ grada postupnim micanjem vremenskog klizača; prometna zagušenja sa jedne strane prometnica (komutiranje prema gradu) te tijekom popodneva zagušenja s druge strane križanja (komutiranje iz grada). Promjenom vremenskog klizača moguće je uočiti promjene u prometnom opterećenju pojedinih prometnica te je također moguće, korištenjem ostalih alata za optimizaciju prometnih tokova, pretpostaviti promjene koje bi se dogodile uvođenjem novih traka, zaobilaznica i drugih vrsta prometnica.

Nakon prolazaka kroz različite korisničke funkcionalnosti aplikacije, u idućem poglavlju biti će riječi o performansama aplikacije te vrsti veze između vremena obrada i različitih atributa vezanih za računalne komponente.

5. TESTIRANJE PERFORMANSI RJEŠENJA

Da bi se sa sigurnošću moglo navesti uvjete i zahtjeve koji pogoduju što bržim vremenima obrade podataka unutar aplikativnog rješenja, potrebno je definirati računalne komponente o kojima aplikativno rješenje ovisi u najvećoj mjeri.

Očekivane komponente koje obavljaju sav teret obrade podataka aplikacije su procesor, radna memorija i tvrdi disk. S tom pretpostavkom na umu, rad aplikacije je testiran u nekoliko različitih okruženja u svrhu saznanja tzv. „uskog grla“; komponente koja svojim performansama najviše utječe na vrijeme obrade podataka.

Prvotna pretpostavka je bila orijentirana na procesorsku obradu podataka kao glavni vremenski faktor; u pitanju su stotine tisuća datoteka. Nadalje, svaka od njih zahtjeva malo manje od 200 iteracija programskog koda da bi podatak iz tekstualne datoteke dospio na svoje hijerarhijsko mjesto unutar ugniježđenih Rječnika i Liste. Sekundarna komponenta čije performanse direktno utječu na vremensku obradu podataka jest radna memorija. Iako je iz teorije i iskustva općepoznato da radna memorija posjeduje veoma visoke brzine prijenosa, čitanja i zapisivanja podataka, brzina odaziva i općenita latencija memorije je uzeta u obzir kao faktor koji može limitirati brzine obrada datoteka. Nadalje, performanse tvrdih diskova su također jedan od ključnih faktora prilikom obrade podataka, kao što je već spomenuto. Njihove mehaničke osobnosti te relativna stara konceptualna zamišljenosti čine ih jednom od komponenata koja posjeduje potencijal biti usko grlo prilikom transfera podataka.

Sa svim informacijama i pretpostavkama definiranimi, u idućim poglavljima riječ će biti o testiranju pojedinih računalnih sustava u vremenima obrade podataka te performansi njihovih komponenata korištenjem programa za mjerenje (engl. *benchmark*). Navedene aktivnosti imaju svrhu nalaženja opipljive veze koja direktno utječe na brzinu obrade podataka unutar aplikativnog rješenja.

5.1. Programi za *benchmark*

Za razliku od nazivnih osobina i tehničkih karakteristika komponenti, iste karakteristike mogu postići veoma veliku vrijednost devijacije u radnim uvjetima. Da bi se veza između performansi pojedinih komponenata računala i vremena obrade podataka mogla odrediti sa što većom točnošću, koristiti će se aplikacije za mjerenje čija je svrha pronalaženje stvarnih atributa i specifikacija. U slučaju mjerenja aplikacije *SegCalc*, koristit će se aplikacije za mjerenje procesorske snage te brzine zapisivanja i čitanja podatka sa radne memorije i tvrdog diska. Dobiveni rezultati usporedit će se sa vremenima obrade na pojedinom računalu te naknadna veza bi trebala pokazati korelaciju između dvije vrijednosti.

LinX [12] je aplikacija za testiranje procesorskih performansi. Korištena verzija je 0.6.5. te je izdana pod *freewave* licencom. *LinX* aplikacija mjeri procesorsku snagu putem rješavanja sustava linearnih jednadžbi u vremenskoj domeni. Također testira stabilnost sustava i otkriva sklopovske probleme. Smatra se jednim od najintenzivnijih alata za testiranje procesora i stabilnosti računalnog sustava. Odlikuje se jednostavnim sučeljem, podrškom za *Intel* te *AMD* marke procesora te 32 i 64-bitnih sustava, čime je pogodan za testiranje svih računala određenih u testnoj skupini (Slika 7).

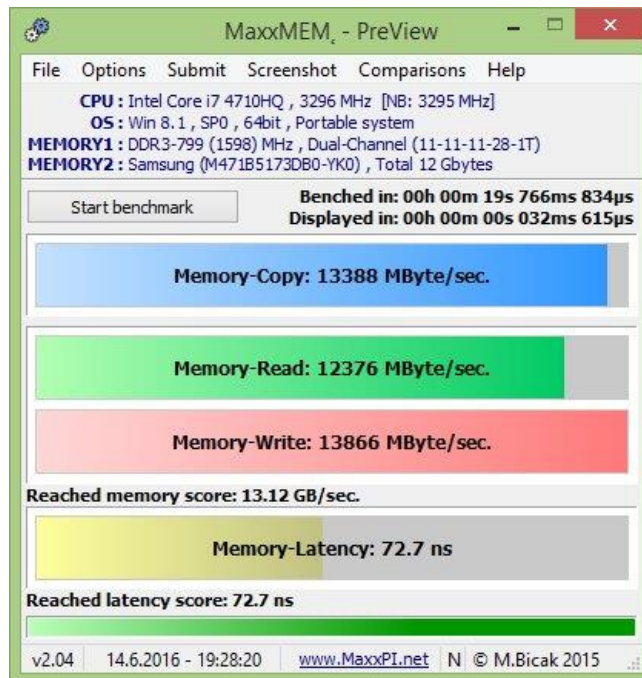
| # | Size | LDA | Align | Time | GFlops | Residual | Residual (norm.) |
|----|-------|-------|-------|--------|---------|---------------|------------------|
| 1 | 10000 | 10008 | 4 | 9.999 | 66.6924 | 9.436774e-011 | 3.327502e-002 |
| 2 | 10000 | 10008 | 4 | 10.166 | 65.5996 | 9.436774e-011 | 3.327502e-002 |
| 3 | 10000 | 10008 | 4 | 10.879 | 61.2994 | 9.436774e-011 | 3.327502e-002 |
| 4 | 10000 | 10008 | 4 | 10.333 | 64.5369 | 9.436774e-011 | 3.327502e-002 |
| 5 | 10000 | 10008 | 4 | 9.357 | 71.2716 | 9.436774e-011 | 3.327502e-002 |
| 6 | 10000 | 10008 | 4 | 9.959 | 66.9623 | 9.436774e-011 | 3.327502e-002 |
| 7 | 10000 | 10008 | 4 | 10.134 | 65.8044 | 9.436774e-011 | 3.327502e-002 |
| 8 | 10000 | 10008 | 4 | 9.482 | 70.3319 | 9.436774e-011 | 3.327502e-002 |
| 9 | 10000 | 10008 | 4 | 9.406 | 70.8968 | 9.436774e-011 | 3.327502e-002 |
| 10 | 10000 | 10008 | 4 | 9.422 | 70.7776 | 9.436774e-011 | 3.327502e-002 |
| 11 | 10000 | 10008 | 4 | 9.372 | 71.1584 | 9.436774e-011 | 3.327502e-002 |
| 12 | 10000 | 10008 | 4 | 9.397 | 70.9669 | 9.436774e-011 | 3.327502e-002 |
| 13 | 10000 | 10008 | 4 | 9.467 | 70.4446 | 9.436774e-011 | 3.327502e-002 |
| 14 | 10000 | 10008 | 4 | 9.402 | 70.9283 | 9.436774e-011 | 3.327502e-002 |

14/20 64-bit 8 threads 71.2716 GFlops peak Intel® Core™ i7-4710HQ

Slika 7. Glavni prozor aplikacije *LinX*

Komponenta koja će se mjeriti u testiranju procesora su FLOPS vrijednosti što označava operacije s „pomičnim zarezom“ po sekundi; rad s decimalnim brojevima (engl. *Floating Operations Per Second*). FLOPS vrijednost je jedna od reprezentativnih vrijednosti performansi procesora jer se ne veže za broj fizičkih i logičkih jezgri, njihovim radnim taktom te radnim taktom sabirnice (engl. *front side bus*) već daje vrijednost koja dolazi kao rezultat rada svih navedenih komponenti zajedno. [13]

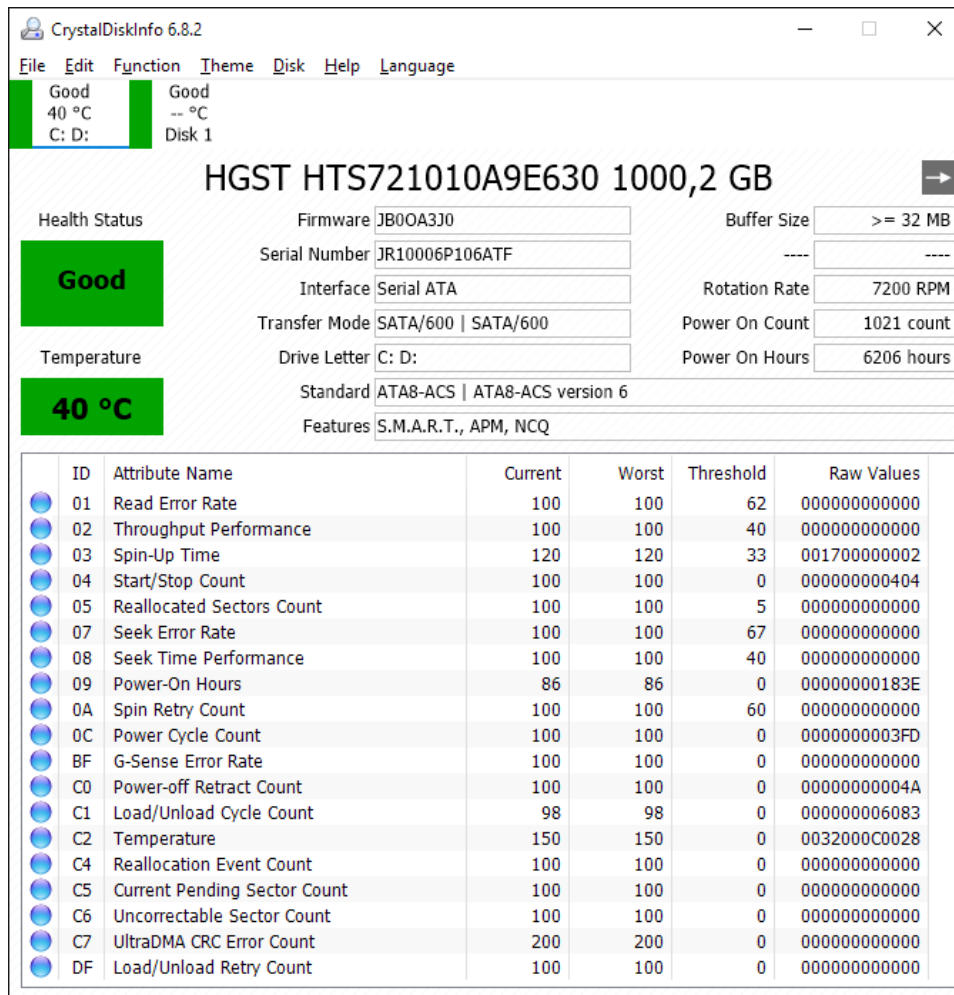
MaxxMem (stilizirana kao *MaxxMem²*) [14] je dio većeg rješenja *MaxxPI²* za mjerenje performansi kompletnog sustava koje određuje performanse radne memorije. Koristi se verzija 2.04. *MaxxMem* se odlikuje veoma jednostavnim sučeljem (slika 8) – unutar prozora nalazi se samo jedan gumb te dovoljno iscrpnim podacima za potrebe testiranja aplikacije *SegCalc*. Sadrži testove koji mjere kopiranje podataka unutar radne memorije, čitanje podataka te klasično zapisivanje. Također se mjeri sveopća propusnost podataka u GB/s te latencija (kašnjenje) najsporijeg dijela podataka u kompletnom testu, nešto što će biti potrebno u kasnijim testiranjima aplikacije.



Slika 8. Glavni prozor aplikacije *MaxxMem*

Određivanje performansi tvrdog diska obavljeno je korištenjem dvije aplikacije istog proizvođača *Crystal Dew World*: *CrystalDiskInfo* te *CrystalDiskMark* [15]. Korištene verzije aplikacija su 6.8.2. odnosno 5.1.2. te su obje preuzete pod *freeware* licencom. Aplikacije slove kao najbolji alati u području dobivanja stvarnih vrijednosti performansa i informacija tvrdog diska.

CrystalDiskInfo (slika 9) je aplikacija koja nudi podatke o tvrdom disku i sučeljima putem kojih se spaja na računalni sustav. Također prikazuje sve podatke vezane za pogreške rada, temperaturu rada i opće stanje tvrdog diska. U radu se aplikacija koristi za potrebe dohvaćanja funkcionalnosti, vrste sučelja te najvažnije; radne brzine okretaja tvrdog diska.

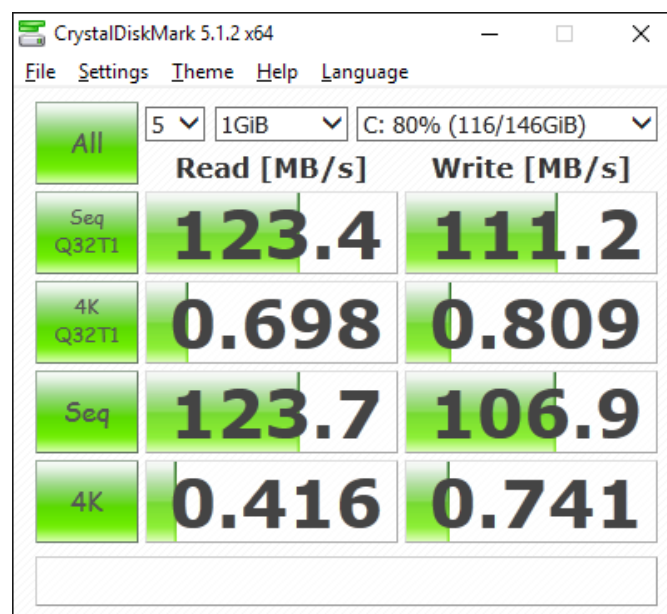


Slika 9. Glavni prozor aplikacije *CrystalDiskInfo*

CrystalDiskMark aplikacija služi za dobivanje brzine čitanja i zapisivanja podataka različitih veličina na tvrdi disk. Sučelje je dizajnirano na logičan način, nudeći testove za čitanje podataka unutar prvog stupca te testove za zapisivanje podataka u drugom, prikazano na slici 10. Moguće je odrediti veličinu uzorka nad kojim će se vršiti testiranje te broj prolaza unutar pojedinačnog testa. Postoje četiri vrste testova:

- Sekvencijalni Q32T1 test,
- 4K Q32T1 test,
- Sekvencijalni test te
- 4K test.

Sekvencijalni testovi mjere čitanje i zapisivanje predodređenih datoteka veličine 128 kilobajta, jednu za drugom. 4K testovi čitaju i zapisuju datoteke veličine 4 kilobajta (4kb), te testovi sa oznakom Q32TX označavaju veličinu međuspremnika (engl. *buffer*), u slučaju programa, za 32 mjesta. X vrijednost označava broj niti (engl. *thread*); paralelnih procesa koji se odvijaju paralelno te se mogu ručno postaviti. Pošto aplikacija *SeqCalc* svoju obradu vrši unutar jedne niti, vrijednosti za potrebe testiranja su postavljene na T1, odnosno testiranje jedne niti obrade. [16]



Slika 10. Glavni prozor aplikacije *CrystalDiskMark*

Test koji je od posebne važnosti za ispitivanja performansi tvrdog diska je 4K Q32T1 iz razloga što testne datoteke veličine 4 kilobajta, u velikoj mjeri odgovaraju datotekama segmenata profila brzina koje su prosjeku veličine 5-7 kilobajta. Osim klasične mjerne jedinice megabajta po sekundi [mb/s], program nudi broj ulazno izlaznih operacija po sekundi nazvanu IOPS vrijednost (engl. *Input Output Operations per Second*). Navedena vrijednost će se koristiti prilikom testiranja tvrdih diskova.

5.2. Računalne platforme korištene prilikom testiranja

U svakidašnjem okruženju, gotovo svako osobno računalo posjeduje drugačiju kombinaciju komponenti. Samim time je teže s točnošću procijeniti samu raznolikost atributa i performansi komponenti koje daju uvjete za ujednačene rezultate.

Tablica 4. Popis testnih računala

| Redni broj | Model procesora | Količina i vrsta memorije | Vrsta diska za spremanje |
|------------|----------------------|---------------------------|--------------------------|
| 1 | AMD X6 FX 6300 | 8GB DDR3 | HDD 7200 RPM |
| 2 | AMD Athlon II X4 620 | 3GB DDR2 | HDD 7200 RPM |
| 3 | Intel Celeron 1000M | 4GB DDR3 | HDD 5400 RPM |
| 4 | Intel Core i5 4200U | 8GB DDR3 | SSD |
| 5 | Intel Core i7 4710HQ | 8GB DDR3 | HDD 7200 RPM |
| 6 | Intel Core i7 4710HQ | 12GB DDR3 | SSD |
| 7 | Intel Core i3 2120 | 4GB DDR3 | SSD |
| 8 | Intel Core i5 3632QM | 8GB DDR3 | SSD |

Za potrebe testiranja aplikativnog rješenja, odabrane su računalne platforme koje će služiti kao osnova testiranju performansi aplikacije. Tablica 4. prikazuje popis računala te bitne specifikacije istih. Prilikom naknadnih referenci računala u rezultatima testiranja, biti će spomenuta po svojem rednom broju zbog razloga jednostavnosti i preglednosti.

Nastojalo se priložiti dovoljno raznolik popis komponenata da bi se korelacija među vremenima obrade i performansama komponenata mogla odrediti s relativnom točnošću.

5.3. Testno okruženje i testni uzorci

Rad aplikacije se testirao s dvije skupine segmenata. Prva skupina je tzv. reprezentativna skupina i sadrži preko dvanaest tisuća segmenata, 166 tisuća pojedinačnih datoteka i zauzima točno 1 GB prostora na tvrdom disku. Druga skupina je čitava skupina segmenata koja se alocirala za potrebe rada te sadrži preko 47 tisuća segmenata i 618 tisuća datoteka. (tablica 5).

Tablica 5. Testne skupine podataka

| Testni uzorci | Veličina | Broj segmenata | Broj datoteka |
|----------------------------------|----------|----------------|---------------|
| Reprezentativni uzorak segmenata | 1,00 GB | 12.858 | 166.558 |
| Kompletni uzorak segmenata | 3,72 GB | 47.689 | 618.878 |

Razlog stvaranja dvije odvojene testne skupine jest sama činjenica da je radna memorija, kao primarna lokacija za spremanje obrađenih podataka aplikacije, ograničeni resurs. Mnoga računala današnjice, pogotovo u nižem cjenovnom rangu, dolaze sa 4 GB radne memorije što je minimum za fluidno funkcioniranje, gledajući sa strane operacijskih sustava te normalne količine instaliranih aplikacija na prosječnom računalu. Kompletni uzorak segmenata zauzima 3,72 GB prostora tvrdog diska te je veoma velika mogućnost da će memorija računala sa samo 4 GB memorije u potpunosti biti iskorištena za vrijeme obrade podataka.

U slučaju popunjavanja radne memorije, Window operacijski sustav pretvara radnu memoriju u virtualnu memoriju. Virtualna memorija je hibridna vrsta memorije koja sadrži svoje podatke na radnoj memoriji i na sekundarnoj lokaciji, najčešće tvrdom disku u slučaju da računalni programi zahtijevaju više memorije od količine koju samo računalo sadrži. Višak memorije se sprema u specifičnu datoteku, tzv. *straničnu datoteku* (engl.

page file) koja se nalazi na tvrdom disku računala, točnije u korijenu C diska (C:/) prigodno nazvanu *Pagefile.sys*.

Mana navedenog procesa jest sama brzina tvrdog diska, koji posjeduje brzina transfera podataka nekoliko puta sporije od radne memorije računala. Pošto neka od računala odabranih za testiranje posjeduju memorije u iznosu od 4 GB, obrada kompletnog uzorka segmenata bi u teoriji trebala trajati značajno duže zbog implementacije spomenutog procesa, nešto o čemu će biti riječi u idućem poglavlju kada će se govoriti o konkretnim rezultatima.

U svrhu zadržavanja reprezentativnosti rezultata testiranja kao i testnog okruženja izdvojena je sekundarna, manja količina podataka veličine 1 GB. Time je osigurano da će sva testna računala moći obraditi podatke i zadržati ih u svojoj radnoj memoriji. Nadalje, računala koja imaju dovoljno veliku količinu radne memorije moći će obaviti obradu većeg dijela segmenata, time pružajući dodatne podatke za statistiku rezultata i njihovo obrazloženje.

5.4. Testiranje

Testiranje aplikacije *SegCalc* se nastojalo odraditi u što kontroliranijim uvjetima. U tu svrhu se testiranje svakog testnog uzorka obavilo tri puta, te se za daljnju obradu uzelo srednje vrijeme obrade. Prije rasprave o rezultatima testiranja, valja napomenuti nekoliko činjenica koje su se istaknule kod samog procesa testiranja.

Tijekom testiranja otkrilo se da antivirusni programi usporavaju obradu podataka od 30 pa čak i do 100 posto. Točnije, moduli antivirusne zaštite koji obavljaju djelatnost zaštite u stvarnom vremenu (engl. *real-time protection*) provjeravaju svaki transfer podataka na računalnu memoriju [17] te se time događa dvostruko otvaranje pojedine datoteke. Dakako, ne zanemaruje se činjenica da će u svakodnevnim situacijama korištenja aplikacije barem neka vrsta antivirusne zaštite biti prisutna.

Budući da različita računala posjeduju drugačije proizvode za antivirusni zaštitu te nemogućnost preciznog određivanja u kojoj mjeri usporavaju testiranje aplikacije, je utjecala na parametre testiranja te testno okruženje. Zbog tih razloga donesena je odluka da se za vrijeme testiranja sve svi moduli zaštite u stvarnom vremenu isključe.

Nadalje, u prvim stadijima testiranja otkrilo se da su naknadni prolazi obrada podataka u pravilu puno brži od inicijalnog prolaza. Naknadno je otkriven mehanizam koji uzrokuje navedeni fenomen. Ispostavlja se da nakon prvog prolaza obrade podataka, podaci koji se zapisuju u radnu memoriju ostaju u njoj nakon što aplikacija bude ugašena. Navedeni mehanizam naziva se *superfetch* i sastavni dio je Microsoftovih operacijskih sustava od objavljivanja Windows 7 OS-a. [18] Windows operacijski sustav čuva često korištene podataka koji se učitavaju sa tvrdog diska u radnu memoriju u slučaju naknadnog korištenja. Navedeni proces kao posljedicu ima visoku razinu korištenja radne memorije, no navedeni podaci posjeduju niski prioritet čuvanja; u slučaju pojavljivanja procesa kojemu su potrebni resursi radne memorije, sačuvani podaci se brišu i zamjenjuju.

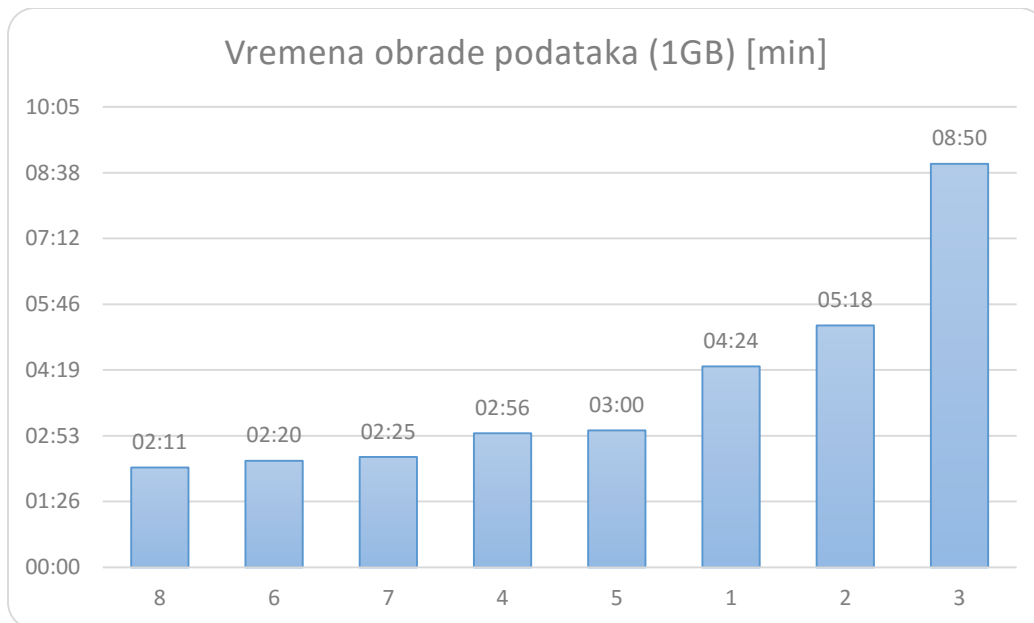
Otkriće navedene činjenice je predstavljalo manje komplikacije prilikom testiranja; nakon svakog testnog prolaza računalo je trebalo biti ugašeno i ponovno pokrenuto. No beneficije ovakvog načina su očite; svako naknadno učitavanje segmenata će biti ubrzano do nekoliko puta prilikom rada aplikacije. Tijekom testiranja zabilježeno je ubrzanje od pet do sedam puta. Navedena činjenica će utjecati na naknadnu diskusiju u sljedećem poglavlju.

5.5. Rezultati testiranja

Testiranje je, kao što je već navedeno, podijeljeno u dvije faze: testiranje referentnog dijela segmenata te testiranje kompletnog seta segmenata. Testiranje referentnog dijela segmenata je obavljeno na svih osam testnih računala, dok je testiranje kompletnog dijela segmenata obavljeno na računalima koja su imala dovoljnu količinu radne memorije, odnosno njih četiri.

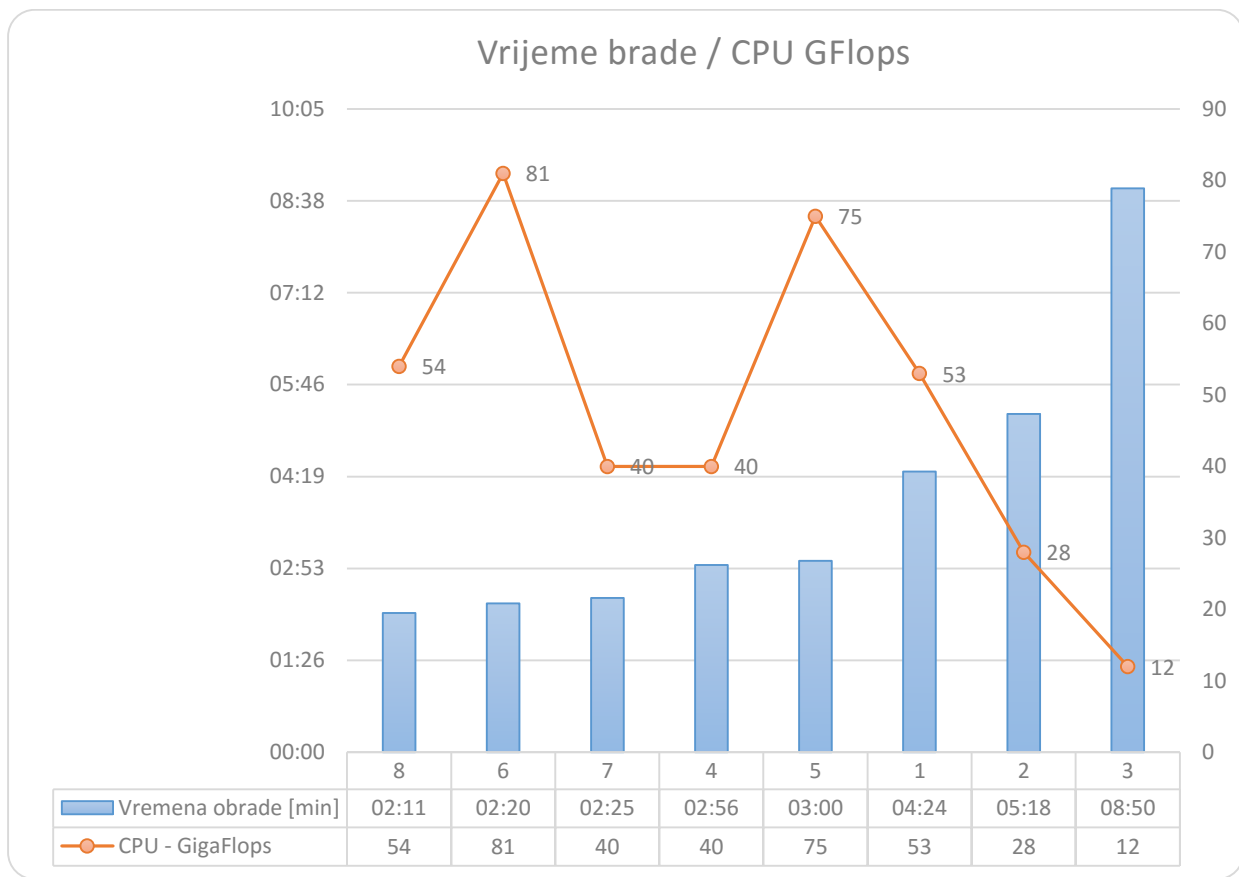
5.5.1. Testiranje referentnog dijela segmenata

Grafikon 1. prikazuje vremena obrade podataka na testnim računalima. Računala se označuju svojim rednim brojem kao što je već napomenuto, te je grafikon posložen da prikazuje vremena obrade od najbržih prema najsporijima.



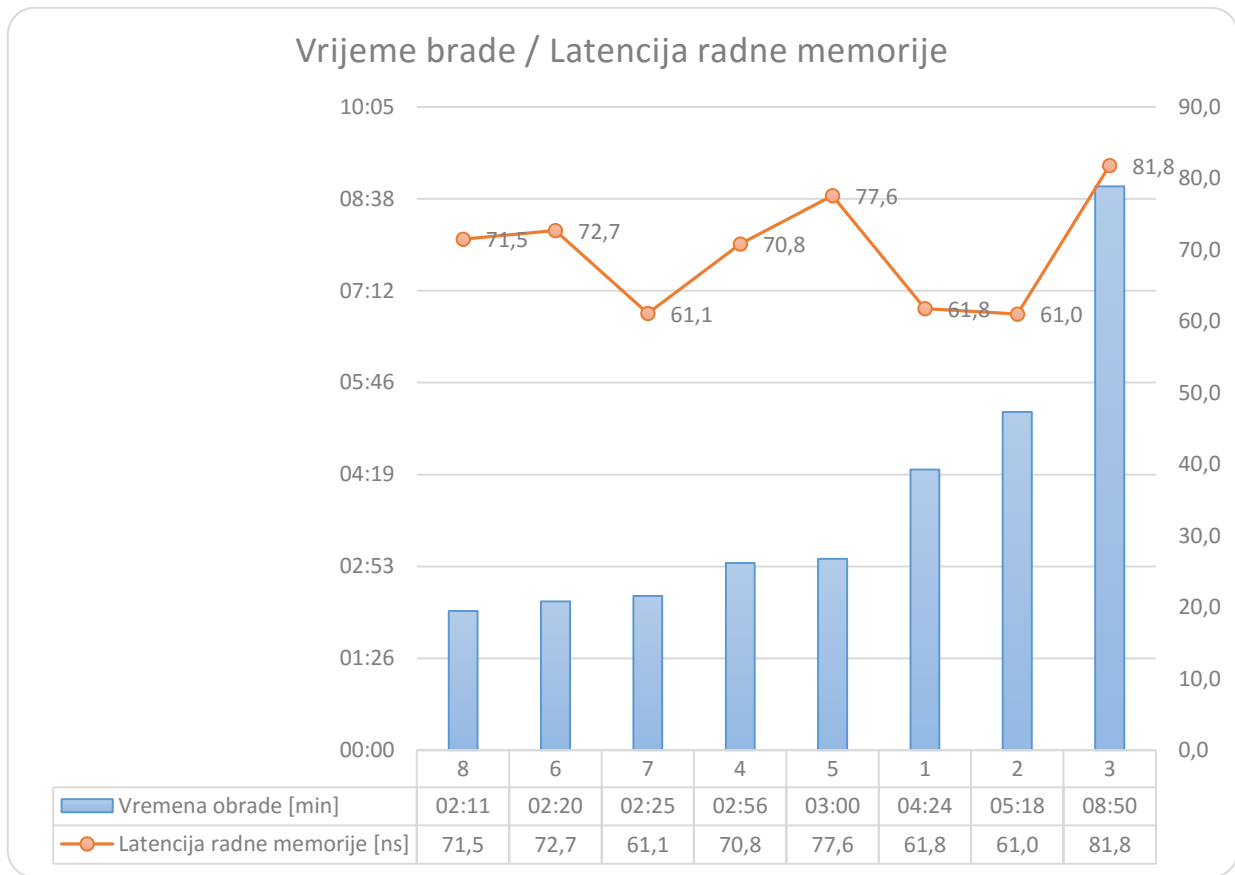
Grafikon 1. Vremena obrade reprezentativnog seta podataka testnih računala

Među prvima, s nakon dvije minute obrade pa nadalje, su se generalno govoreći smjestila računala s boljim performansama te su sporija vremena obrade imala računala sa slabijim komponentama. Naravno, takve okvirne pozicije računala su bile očekivane, no daljnji grafikon pokazuje izravnu vezu određenih komponentata s vremenima obrade. Grafikon 1. će poslužiti kao referentni okvir prilikom usporedbe performansi relevantnih komponentata računalnih sustava.



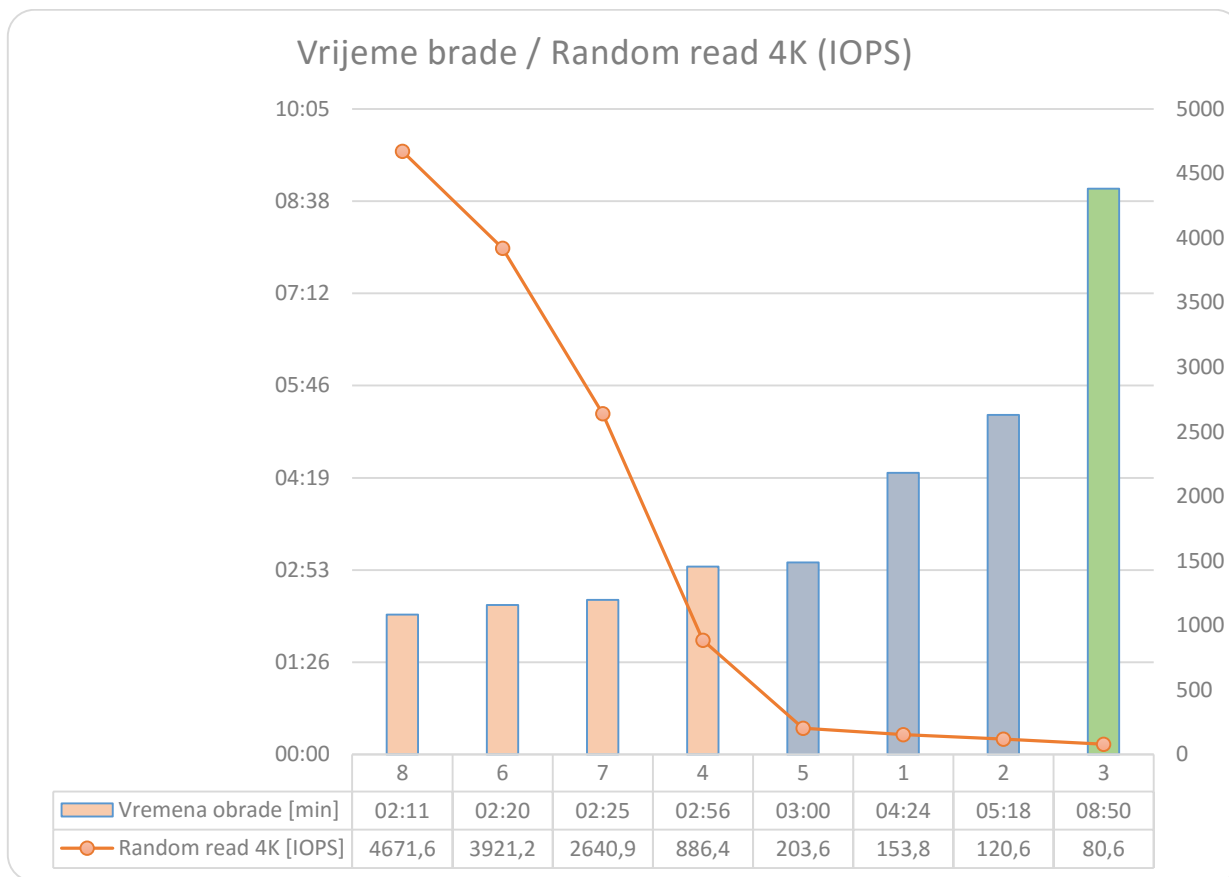
Grafikon 2. Vrijeme obrade podataka u odnosu na *Flops* vrijednosti procesora

Testni rezultati procesorske snage pokazuju određenu korelaciju koja određuje vrijeme obrade, no nikako nije dovoljno dosljedna da se može reći da direktno utječe na navedena vremena, kao što pokazuje grafikon 2. Zapravo, u većini slučajeva procesorska iskoristivost nije zabilježena iznad 70% za vrijeme obrade podatka unutar aplikacije *SegCalc*, što daje za zaključiti da druga komponenta koči više razine iskoristivosti procesora.



Grafikon 3. Vrijednosti obrade podataka u odnosu na latenciju radne memorije

Analizirajući rezultate dane grafikonom 3., može se zaključiti da su vrijednosti latencija radne memorije testnih računala približno iste. Razlika od 20 nanosekundi naizgled nema gotovo nikakav utjecaj na vremena obrade podatka. Kao što je prikazano na slici 4. prethodnog poglavlja, brzine čitanja i zapisivanja podataka unutar radne memorije daleko prelaze slične brzine zabilježene kod tvrdih diskova.



Grafikon 4. Vremena obrade podataka u odnosu na IOPS vrijednosti tvrdih diskova

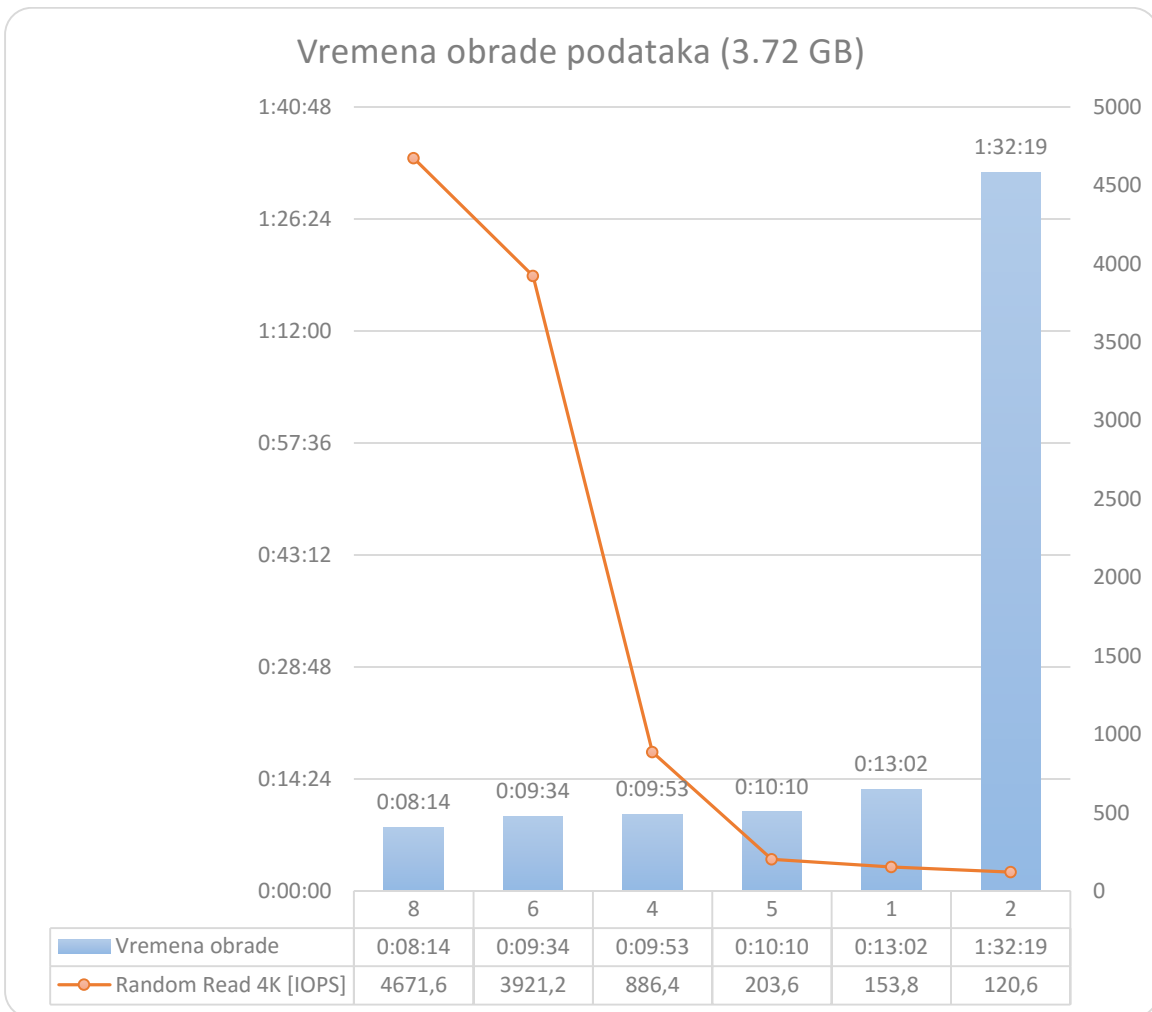
U slučaju rezultata testiranja tvrdih diskova testnih računala, došlo je do zanimljivih otkrića. Grafikon 4. koji pokazuje ovisnost vremena obrade o IOPS vrijednosti, koja pokazuje maksimalan broj operacija čitanja po sekundi, otkriva jasnu vezu. Brzina čitanja datoteka sa tvrdog diska direktno utječe na vrijeme obrade. Računala koja su imala najveću IOPS vrijednost također su obradila podatke u najkraćem vremenu te vrijedi i obrnuto. Također, vrijedi da vrsta tvrdog diska znatno utječe na vrijeme obrade. Računala 8, 6, 7 i 4 koriste SSD disk, računala 5, 1, 2 koriste tvrdi disk s brzinom od 7200 okretaja u minuti te tvrdi disk računala 3 se okreće 5400 puta u minuti. Zanimljivo opažanje jest da iako SSD diskovi posjeduju do nekoliko desetaka puta veću brzinu čitanja datoteka, brzina obrade podataka je tek 10-25% kraća nego ona od klasičnih tvrdih diskova. Dodatna istraživanja su potrebna za utvrđivanje uzroka navedene nedosljednosti.

Iz prikazanoga je moguće uočiti da se 4K Q32T1 test pokazao veoma iscrpljujućim u slučaju tvrdih diskova. Svi klasični tvrdi diskovi su pokazali veoma male brzine prijenosa u odnosu na brzine čitanja velikih datoteka, što je potvrdilo pretpostavku da je čitanje veoma mnogo malih datoteka najsporija operacija u terminima količine pročitanih datoteka u jedinici vremena.

5.5.2. Testiranje kompletnog seta segmenata

Gledajući rezultate vremena obrade podataka kompletnog dijela segmenata, moguće je jednako zaključiti, kao kod testne skupine, da je tvrdi disk glavna komponenta o kojoj ovisi vrijeme obrade. Također, odnos povećanja vremena obrade s povećanjem ukupne količine podataka je linearan.

Također, gledajući grafikon 5., vrijedi obratiti posebnu pozornost na rezultate računala 2, koje može poslužiti kao primjer korištenja stranične datoteke; spremanja podataka radne memorije na tvrdi disk. Tijekom testiranja, radna memorije računala 2 se u potpunosti popunila. U tom trenutku, računalo je krenulo spremati podatke na tvrdi disk što ga je, uz naknadno te simultano čitanje podataka za potrebe aplikacije, dodatno opteretilo. Dakle podaci koji se koriste za obradu se čitaju sa tvrdog diska, te nakon obrade se također zapisuju na tvrdi disk; mehanika koja bi se poistovjećivala sa radom lokalne baze podataka.



Grafikon 5. Vremena obrade svih segmenata u odnosu na IOPS vrijednost

Rezultat navedenih mehanizma je izrazito dugačko vrijeme obrade. Promatrajući dugo vrijeme obrade računala 2, moguće je vizualizirati kako bi otprilike izgledalo vrijeme obrade da je aplikacija dizajnirana sa lokalnom bazom podataka u koju sprema podatke.

5.5.3. Zaključci testiranja

Originalna pretpostavka da je proces obrade podataka brži prilikom spremanja podataka u radnu memoriju od načina spremanja u bazu podataka je još uvijek istinita. U slučaju spremanja podataka u radnu memoriju, vrijeme obrade će uvijek biti isto; dapače

čak i brže ukoliko se aplikacija ne gasi te podaci ostaju unutar radne memorije za vrijeme radnih termina korištenjem *superfetch* mehanizma. Suprotno tome, korištenje baze podataka prilikom obrade vrijednosti segmenata bi imalo za posljedicu puno duža vremena učitavanja podataka. To je zato jer bi se u isto vrijeme odvijala dva procesa: čitanje segmenata sa tvrdog diska te spremanje istih također na tvrdi disk u lokalnu bazu podataka. Dva takva procesa simultanog čitanja i zapisivanja velikog broja datoteka i vrijednosti bi u teoriji produžilo vrijeme učitavanja za minimalno 80% te bi svako naknadno učitavanje trajalo jednako kao dugo. U tom slučaju, vrijednosti ne bi ostajale privremeno pohranjene unutar radne memorije i spremne za ponovno korištenje kao kod trenutnog korištenja što je mala, ali nikako neznatna činjenica.

Dakako postoje mane spremanja vrijednosti u radnu memoriju, no one su prilično očite – privremeno pohranjivanje te mali kapacitet. Kao što je rečeno u prijašnjim poglavljima, podaci ostaju spremljeni te spremni za korištenje samo za vrijeme rada aplikacije, te iako ostaju privremeno spremljeni kao privremeno mjesto spremanja (*cache*) nakon što aplikacija prekine sa radom, takva vrsta podataka je podložna naknadnom prepisivanju drugih podataka preko vrijednosti segmenata, te bi ponovno učitavanje istog broja segmenata trajalo jednako kao i prvo originalno učitavanje. Također, sekundarna mana jest gornja granica takvog načina pohrane, što je kapacitet radne memorije, nešto što je različito od računala do računala. Na računalima koja posjeduju velike kapacitete radne memorije (primjerice serverska računala te računala višeg cjenovnog ranga) teoretski je izvediv ovakav način obrade i zapisivanja podataka. Jedan gigabajt segmenata prilikom učitavanja zauzima od 600 do 650 megabajta unutar radne memorije; razlozi tome su spremanje samo određenih stupaca (vrijeme i apsolutna brzina) te učinkovitiji način spremanja varijabli unutar već spomenutih struktura. Ako se mogu povući paralele između količine podataka i geografskog područja unutar kojeg se segmenti iz odabrane količine podataka nalaze, računalo sa 16 gigabajta radne memorije bi moglo učitati i „držati“ u jednom trenutku područje veličine 10 gradova Zagreba. Takva veličina implicira broj od oko 500.000 segmenata. Što može biti sasvim dovoljno, ali i ne mora biti; činjenica koja naravno ovisi o potrebama pregleda i širine geografskog područja koju je potrebno obraditi.

Sljedeće poglavlje sadrži diskusiju o napretku aplikacije, njezinih potencijalnih varijanata, profila brzina te načinu manipulacije podataka unutar vremena obrade.

6. PREPORUKE I SMJERNICE ZA DALJNI NAPREDAK

Zaključak koji se može izvesti iz prethodnog teksta jest izbor između dva načina izrade. Aplikacija napravljena, korištena i testirana za potrebe ovo rada odlikuje se bržim vremenom inicijalne obrade podataka te nevezanosti za lokalnu bazu podataka što implicira mobilnost. Sa druge strane njene mane su privremenost obrađenih podataka te limitirani kapaciteti količine podataka koji se mogu obraditi. Autor smatra da je aplikacija pogodna za primjenu na manjim regionalnim područjima kao što su područja velikih gradova te ruralne sredine većeg geografskog područja koje je slabije povezano i analogno time posjeduje manji broj segmenata po četvornom kilometru.

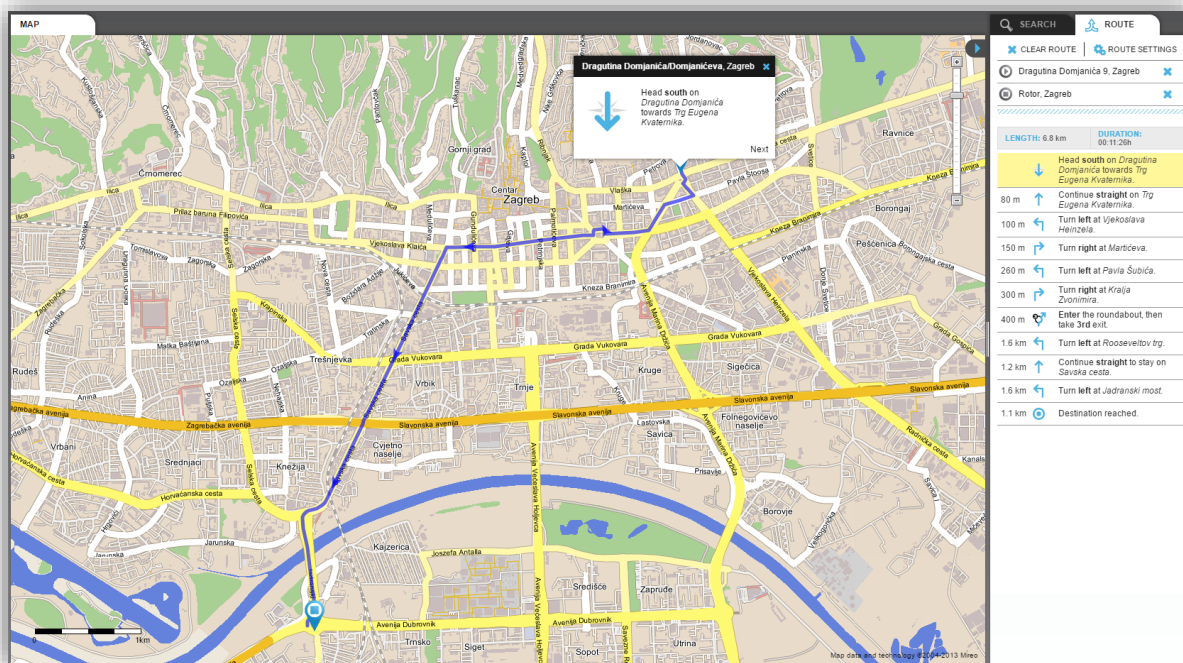
Što se tiče aplikacije *SegCalc*, napredak u smislu učinkovitijeg provođenja koda postoji. Iako je sam kod optimiziran za operacije obrade, činjenica je da iskorištava samo jednu procesorsku nit [19]. Procesorska nit (engl. *thread*) je termin koji označava izvođenje samo jedne male skupine izvršavanja te je moguće imati više niti koje se izvode paralelno i pridonose performansama programa. Sa time je moguće da svaka logička jezgra procesora može podržati nekoliko paralelnih niti koje dijele njezine resurse i privremenu memoriju. Kada se takav proces pomnoži sa brojem logičkih jezgara (osam u slučaju računala na kojemu autor trenutno piše te do 72 jezgre u slučaju trenutno vodećeg *Intel*-ovog procesora [20]) razlika je prilično osjetna. Takvo poboljšanje bi se posebice očitalo u grafičkom dijelu programa, gdje bi se prikazivanje segmenata u dvodimenzionalnom prostoru i njihovo dinamičko mijenjanje boja moglo ubrzati. Također bi se korištenjem naprednijeg i kompleksnijeg koda za prikazivanje segmenata moglo uključiti više mogućnosti za vizualizaciju, odabir i manipulaciju segmentima. Upotreba i konstruiranje upravljačkih potprograma i mehanizma za upravljanjem više procesorskih niti je veoma kompleksan zadatak; nešto što se koristi prilikom izrade kompleksnih programa obrade, igara te operacijskih sustava.

U pogledu vizualnog elementa prikaza segmenata, postoji veoma veliki prostor za nadogradnju. Aplikacija *SegCalc* koristi primitivan grafički sustav koji crta linije povezane točkama što predstavljaju krajnje točke koordinata segmenata. Podloga

Prikaz je limitiran unaprijed odabranim i učitanim segmentima te u slučaju naknadnog odabira, prvotna skupina obrađenih podatka se briše.

Napredak razvoja aplikativnog rješenja pa i cijela filozofija iza razvoja aplikacije može se promatrati sa strane već razvijenih rješenja koja nude sličnu funkcionalnost.

Mireo d.d. nudi svoju aplikaciju za GPS navigaciju i praćenje koristeći identične profile brzina kao i ovaj rad. Njihovo rješenje koristi navedene podatke za predikciju prometnih opterećenja te na temelju toga u stvarnom vremenu dinamički mijenja rutu u svrhu maksimalno skraćenog vremena putovanja, prikazano slikom 11.



Slika 11. Sučelje web-bazirane navigacije tvrtke *Mireo d.d.*

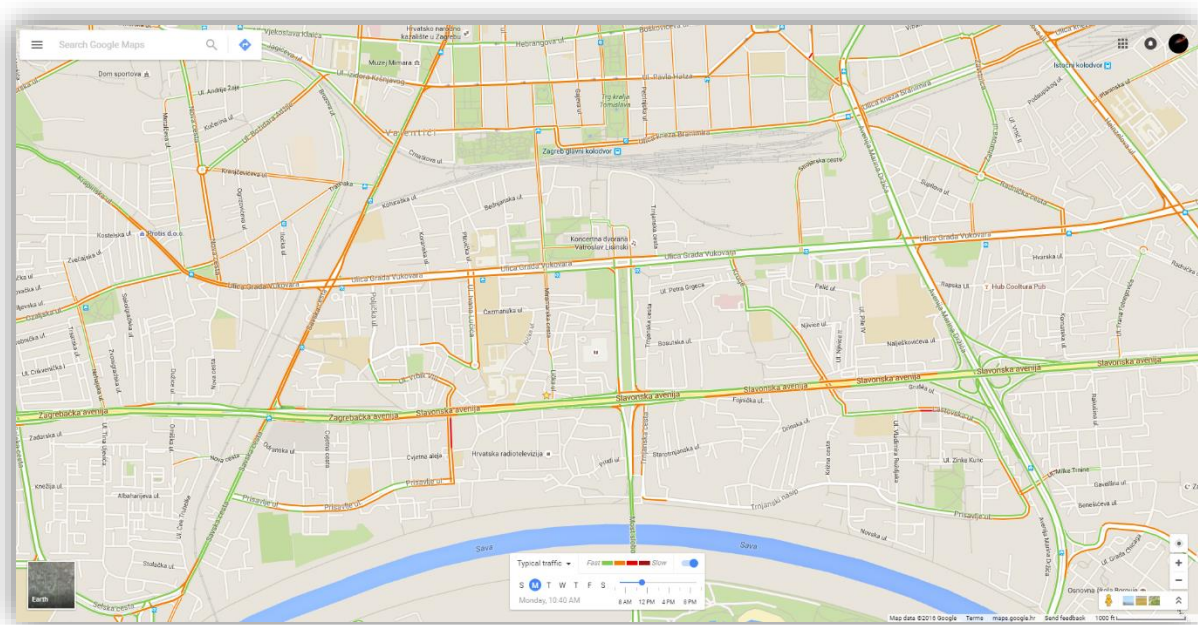
Izvor: [23]

Karta se odlikuje izvrsnim vremenima odaziva te veoma malim vremenima čekanja prilikom izračuna rute. Nažalost, aplikacija ne pokazuje prometna opterećenja na jednom od slojeva karte, već samo koristi svoje algoritme navigacije zasnovane na njima. Također

se mobilne verzije karti naplaćuju. No, prednost ove aplikacije je u tome što posjeduje slike karata koje se poklapaju sa segmentima prometnica, time olakšavajući snalaženje u prostoru korištenjem aplikacije. Osim što je aplikaciju moguće koristiti preko Interneta, *Mireo d.d.* također nudi varijante za *Android*, *iOS* te *BlackBerry* operacijske sustave te se naplaćuju, kao što je spomenuto.

Drugi primjer korištenja prometnih opterećenja u svrhe navigacije jest *Google Maps*. Osim što nudi topološke karte te karte satelita, moguće je korištenje tzv. prometnog sloja, načina pregledavanja koji prikazuje prometna opterećenja poput aplikacije *SegCalc*.

Web-bazirana aplikacija nudi pregršt mogućnosti koje su veoma slične aplikaciji izrađenoj u ovom radu. Moguće je pretraživanje prometa po danima u tjednu te satima u danu. Ono što *Google Maps* ne prikazuje su profili brzina te specifične brzine u određenom vremenskom intervalu. Prometno zagušenje je moguće razaznati samo putem vizualno obojanih dijelova prometnica, kao što je prikazano na slici 12.



Slika 12. sučelje lokacijskog sustava *Google Maps*-a

Izvor: [22]

No, navedeni „propusti“ nisu od presudne važnosti za prosječnog korisnika te se aplikacija u cijelosti smatra veoma korisnim alatom zbog svoje besplatne prirode te prvenstveno zbog mogućnosti pregleda prometnih opterećenja u stvarnom pogledu. Kao što je poznato, uređaji sa *Android* operacijskim sustavom su jedni od najzastupljenijih vrsta uređaja na svijetu, te Google kao vlasnik može dohvatiti lokacijske podatke iz svakog uređaja koji ima uključene lokacijske servise. Navedena činjenica znači da u praksi velike arhivske količine podataka nisu potrebne prilikom određivanja prometnih opterećenja jer postoje milijuni korisnika koji voze vozila te time postaju živa statistika na licu mjesta.

Navedena rješenja posjeduju gotovo apsolutnu superiornost prilikom obrade i prikazivanja prostornih podataka što i nije čudo uzimajući u obzir količinu financijskih i intelektualnih resursa posjeduju tvrtke koje su ih izradile. Moguće je zaključiti da aplikativno rješenje *SegCalc* nudi zadovoljavajuće razine korisničkog iskustva i funkcionalnosti sa obzirom na sredstva i podatke koje koristi. S time na umu, uvijek je moguće inkorporirati navedene napredne mogućnosti promatrajući navedena rješenja. Pitanje je samo truda, vremena te resursa potrebnih za ostvarenje funkcionalnosti koje su potrebne ciljanoj skupini korisnika aplikacije *SegCalc*.

U zaključku će biti priložen sažetak sadržaja diplomskog rada te finalne misli o izvršenom trudu i obavljenom testiranju.

7. ZAKLJUČAK

Informacije, otkrića i saznanja samo su jedni od mnogih pojmova koji se asociraju s interpretacijom velikih setova podataka. U svakom slučaju, kvaliteta i rezolucija rezultata linearno raste porastom broja uzoraka i izvornih podataka. No samim time je linearno teže i kompleksnije izvesti obradu navedenih podataka.

Za potrebe diplomskog rada izrađena je aplikacija za obradu navedenih prostornih podataka te je opisan postupak pretvorbe podataka u format spreman za vizualno prikazivanje. Način obrade te lokacija zapisivanja i korištenja podataka su zanimljivi atributi aplikacije koji posebice mogu pomoći pri smanjenju vremena obrade u određenim uvjetima.

Rad aplikacije se smatra da je na zadovoljavajućoj razini. Iako uvijek postoji mjesta za napredak i poboljšanje, mišljenje autora je da je ovakva „ogledna“ verzija aplikacije dovoljan primjer učinkovitosti alternativnih načina obrade navedenih setova podataka.

Rezultati testiranja aplikacije definirali su nekoliko važnih činjenica. Vrijeme obrade podataka direktno ovisi o performansama glavnog medija za pohranu, tvrdog diska. Imajući navedenu činjenicu na umu, preporuča se korištenje SSD diska što većih performansi za postizanje kraćih vremena obrade te brzini prikazivanja segmenata.

Nadalje, mogućnosti proširenja funkcionalnosti su opisane te uspoređene u odnosu na postojeća rješenja. Daljnji razvitak aplikacije će ovisiti o budućim potrebama korisnika. Razvoj budućih sistema te razvitak performansi tvrdih diskova te ostalih komponenata znatno će optimizirati buduće iteracije aplikativnog rješenja.

POPIS LITERATURE

[1] *Visual Studio Community*

Internet stranica: <https://www.visualstudio.com/en-us/products/visual-studio-community-vs.aspx> (lipanj, 2016.)

[2] *Galileo programme*

Internet stranica: <http://www.gsa.europa.eu/galileo/programme> (svibanj, 2016.)

[3] Lazarević, Ž.: *Primena GPS tehnologije*

Internet stranica:

https://www.academia.edu/8351372/Seminarski_rad_Primena_GPS_tehnologije (lipanj, 2016.)

[4] *What exactly is NMEA data?*

Internet stranica: <http://gpsworld.com/what-exactly-is-gps-nmea-data/> (svibanj, 2016.)

[5] Urbano, F., Basille, M.: *Spatial Database for GPS Wildlife Tracking Data*, Springer, Italija, 2014.

[6] Czerniak, R. J.: *Collecting, Processing and Integrating GPS Data into GIS*, National Academy Press, Washington D.C., 2002.

[7] Markham, Paula T.: *Effectiveness of a Program Using a Vehicle Tracking System, Incentives, and Disincentives to Reduce the Speeding Behavior of Drivers With ADHD*, Journal of Attention Disorders, Vancouver, SAD, 2011.

[8] *Mireo DON'T PANIC GPS navigacija*

Internet stranica: <http://www.mireo.hr/hr/gps-navigacija> (lipanj, 2016.)

[9] *Cubic smoothing spline*

Internet stranica:

<http://www.mathworks.com/help/curvefit/csaps.html?requestedDomain=www.mathworks.com> (svibanj, 2016.)

[10] *Dictionary <TKey, TValue> Class*

Internet stranica: [https://msdn.microsoft.com/en-us/library/xfhwa508\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/xfhwa508(v=vs.110).aspx) (travanj, 2016.)

[11] *IndexOutOfRangeException Class*

Internet stranica: [https://msdn.microsoft.com/en-us/library/system.indexoutofrangeexception\(v=vs.110\).aspx](https://msdn.microsoft.com/en-us/library/system.indexoutofrangeexception(v=vs.110).aspx) (travanj, 2016.)

[12] *LinX 0.6.5.*

Internet stranica: <http://linx.en.lo4d.com/> (svibanj, 2016.)

[13] *GFlops, G-hours, and CPU hours*

Internet stranica:

http://www.gridrepublic.org/joomla/components/com_mambowiki/index.php/GFlops,_G-hours,_and_CPU_hours (svibanj, 2016.)

[14] *Download (MaxxMEM² - PreView)*

Internet stranica: <http://www.maxxpi.net/pages/downloads/maxxmemsup2---preview.php> (svibanj, 2016.)

[15] *Crystal Dew World*

Internet stranica: <http://crystalmark.info/?lang=en> (svibanj, 2016.)

[16] *Using Microsoft DiskSpd to Test Your Storage Subsystem*

Internet stranica: <http://sqlperformance.com/2015/08/io-subsystem/diskspd-test-storage> (lipanj, 2016.)

[17] *Windows Defender real-time protection*

Internet stranica: <http://windows.microsoft.com/en-us/windows/defender-realtime-protection-on-off#1TC=windows-7> (lipanj, 2016.)

[18] *Inside the Windows Kernel: Part 2*

Internet stranica: <https://technet.microsoft.com/en-us/magazine/2007.03.vistakernel.aspx> (lipanj, 2016.)

[19] *What are threads, and what do they do in the processor?*

Internet stranica: <http://superuser.com/questions/329904/what-are-threads-and-what-do-they-do-in-the-processor> (lipanj, 2016.)

[20] *Intel unveils 72-core x86 Knights Landing CPU for exascale supercomputing*

Internet stranica: <http://www.extremetech.com/extreme/171678-intel-unveils-72-core-x86-knights-landing-cpu-for-exascale-supercomputing> (lipanj, 2016.)

[21] Erdelić, T.: *Sustav za optimizaciju ruta u dinamičkom transportnom okruženju*, Fakultet prometnih znanosti, Sveučilišta u Zagrebu, 2014.

[22] *Google Maps*

Internet stranica: <https://www.google.com> (lipanj, 2016.)

[23] *Mireo Maps*

Internet stranica: <http://maps.mireo.hr/GelinX/> (lipanj, 2016.)

POPIS AKRONIMA I KRATICA

| Akronim ili kratica | Broj Stranice | Prijevod ili opis akronima ili kratice |
|---------------------|---------------|--|
| GPS | 3 | engl. <i>global positioning system</i> (globalni pozicijski sustav) |
| GLONASS | 4 | rus. <i>globalnaya navigatsionnaya sputnikovaya sistema</i> (gobalni navigacijski satelitski sustav) |
| GALILEO | 4 | europski satelitski sustav |
| NMEA | 5 | engl. <i>national marine electronic association</i> (nacionana pomorska elektronička asocijacija) |
| RTK | 5 | engl. <i>real time kinematics</i> (kinematika u stvarnom vremenu) |
| HDOP | 6 | engl. <i>horizontal dilution of precision</i> (horizontalno smanjenje preciznosti) |
| GIS | 7 | engl. <i>geographic information system</i> (geografski informacijski sustavi) |
| ADHD | 7 | engl. <i>attention deficit hyperactivity disorder</i> (poremećaj hiperaktivnosti i deficita pažnje) |
| SORDITO | 8 | Sustav za optimizaciju ruta u dinamičkom transportnom okruženju |
| GUI | 12 | engl. <i>graphic user interface</i> (grafičko korisničko sučelje) |
| SQL | 12 | engl. <i>structured query language</i> (strukturirani jezik upita) |
| SSD | 13 | engl. <i>solid state drive</i> |
| RAM | 14 | engl. <i>random accessed memory</i> (radna memorija) |
| FLOPS | 28 | engl. <i>floating operations per secong</i> (operacije sa decimalnim brojevima po sekundi) |
| IOPS | 31 | engl. <i>input-output operations per seond</i> (ulazno-izlazne operacije po sekundi) |

POPIS STRANIH IZRAZA

| Strani izraz | Broj stranice | Prijevod ili opis stranog izraza |
|-------------------------|---------------|--|
| line of sight | 4 | optička linija |
| checksum | 6 | kontrolna suma |
| free flow interval | 8 | interval slobodnog toka |
| cubic smoothing spline | 10 | vrsta krivulje koja zaokružuje međusobne vrijednosti |
| Winforms, Windows Forms | 12 | Windows forme |
| dictionary | 13 | rječnik |
| list | 14 | lista |
| array | 14 | polje |
| int | 14 | cjelobrojna vrijednost |
| double | 14 | vrijednost dvostruke preciznosti |
| string | 14 | znakovna vrijednost |
| datetime | 14 | vrijednost datuma i vremena |
| string array | 15 | polje znakovnih vrijednosti |
| tab | 15 | tabulator |
| .NET Framework | 18 | .NET okviri |
| progress bar | 19 | traka za napredak |
| slider | 20 | klizač |
| trackbar control | 20 | kontrola klizača |
| benchmark | 26 | mjerilo |
| freeware | 27 | vrsta softvera koja je besplatna za distribuciju |
| front side bus | 28 | sabirnica |
| buffer | 31 | međuspremnik |
| processing thread | 31 | procesirajuća nit |
| page file | 34 | stranična datoteka |

| | | |
|----------------------|----|--|
| real-time protection | 34 | zaštita u stvarnom vremenu |
| superfetch | 35 | mehanika privremenog spremanja često korištenih datoteka |
| cache | 42 | privremena lokacija za spremanje vrijednosti |

POPIS ILUSTRACIJA

Popis slika

| | |
|---|----|
| Slika 1. Odabrano područje grada Zagreba kao izvor profila brzina | 9 |
| Slika 2. Uvodni program aplikativnog rješenja SegCalc..... | 19 |
| Slika 3. Glavni prozor aplikacije sa naglašenim kontrolama..... | 20 |
| Slika 4. glavni prozor aplikativnog rješenja SegCalc..... | 21 |
| Slika 5. Grafikon dvosmjernih profila brzina sa jasno prikazanim vremenima zagušenja unutar aplikacije SegCalc | 23 |
| Slika 6. Prikaz zagušenja centra grada Zagreba..... | 24 |
| Slika 7. Glavni prozor aplikacije LinX | 28 |
| Slika 8. Glavni prozor aplikacije MaxxMem..... | 29 |
| Slika 9. Glavni prozor aplikacije CrystalDiskInfo | 30 |
| Slika 10. Glavni prozor aplikacije CrystalDiskMark | 31 |
| Slika 11. Sučelje web-bazirane navigacije tvrtke Mireo d.d..... | 45 |
| Slika 12. sučelje lokacijskog sustava Google Maps-a..... | 46 |

Popis tablica

| | |
|---|----|
| Tablica 1. Primjer datoteke sa profilima brzina | 10 |
| Tablica 2. Proces pretvorbe „sirovih“ podataka u Listu | 16 |
| Tablica 3. Hijerarhijska struktura podataka | 17 |
| Tablica 4. Popis testnih računala | 32 |
| Tablica 5. Testne skupine podataka | 33 |

Popis grafikona

| | |
|---|----|
| Grafikon 1. Vremena obrade reprezentativnog seta podataka testnih računala | 36 |
| Grafikon 2. Vrijeme obrade podataka u odnosu na Flops vrijednosti procesora..... | 37 |
| Grafikon 3. Vrijednosti obrade podataka u odnosu na latenciju radne memorije..... | 38 |
| Grafikon 4. Vremena obrade podataka u odnosu na IOPS vrijednosti tvrdih diskova .. | 39 |
| Grafikon 5. Vremena obrade svih segmenata u odnosu na IOPS vrijednost..... | 41 |

Popis jednadžbi

| | |
|-----------|----|
| [1] | 10 |
|-----------|----|

POPIS PRILOGA

U sklopu pisanja ovo rada razvijeno je programsko rješenje *SegCalc*, koje se nalazi na optičkom mediju priloženom na zadnjoj korici rada. Prije pokretanja, potrebno ga je instalirati na računalu (otvoriti datoteku Setup.exe).

Detaljne upute za korištenje nalaze se u poglavlju 4.

METAPODACI

Naslov rada: Ekstrakcija informacija o prometnim zagušenjima iz velikih baza podataka GPS-om praćenih vozila

Student: Roman Tušek

Mentor: izv. prof. dr. sc. Tonči Carić

Naslov na drugom jeziku (engleski): Extraction of traffic congestion information from large databases of GPS tracked vehicles

Povjerenstvo za obranu:

- dr. sc. Ivan Grgurević predsjednik
- izv. prof. dr. sc. Tonči Carić mentor
- dr.sc. Pero Škorput član
- dr. sc. Hrvoje Gold zamjena

Ustanova koja je dodijelila akademski stupanj: Fakultet prometnih znanosti Sveučilišta u Zagrebu

Zavod: Zavod za informacijsko komunikacijski promet

Vrsta studija: diplomski studij

Studij: Informacijsko Komunikacijski Promet (npr. Promet, ITS i logistika, Aeronautika)

Datum obrane diplomskog rada: 5. srpnja 2016.

Napomena: pod datum obrane diplomskog rada navodi se prvi definirani datum roka obrane.

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ diplomski rad

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz

necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj

visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu

objavu

_____ diplomskog rada

pod

naslovom

Ekstrakcija informacija o prometnim zagušenjima iz velikih

baza podataka GPS-om praćenih vozila

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom

repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, _____ 24.6.2016. _____

_____ (potpis)