

Univerzitet u Beogradu

Matematički fakultet

**Paralelizacija i programska implementacija  
algoritma za izračunavanje mapa  
gravitacionih sočiva**

**Tatjana Slavko Jakšić**

Master rad

Oblast: Astroinformatika

Beograd, 2010

Ova teza je urađena na Matematičkom institutu Srpske akademije nauka i umetnosti u Beogradu u okviru projekta "Reprezentacije logičkih struktura i primene u računarstvu". Njenom izradom je rukovodio prof. dr Žarko Mijajlović, redovni profesor Matematičkog fakulteta Univerziteta u Beogradu, a komentor je bio dr Predrag Jovanović, naučni saradnik Astronomske opservatorije u Beogradu, kojima se zahvaljujem na korisnim primedbama i sugestijama koje su doprinele da ovo izlaganje bude preciznije i metodološki bolje. Takođe se zahvaljujem prof. dr Nadi Pejović, vandrednom profesoru Matematičkog fakulteta Univerziteta u Beogradu, na trudu uloženom prilikom redigovanja ove teze kao i dr. Luki Popoviću naučnom saradniku Astronomske opservatorije u Beogradu na korisnim sugestijama.

Zahvaljujem se i kolegama sa Matematičkog instituta, Bojanu Marinkoviću, Sanji Roksandić, Aleksandru Pejoviću, Đorđu Baraliću i Branislavu Šimpragi na pomoći u prevodu i boljem razumevanju tehničkih aspekata u oblasti paralelnog programiranja. Takođe se zahvaljujem Antunu Balažu i Vladimiru Slavniću sa Instituta za Fiziku u Beogradu, koji su mi omogućili da paralelni program testiram na većem broju procesora.

Rad na temi paralelizacije je bio izuzetno interesantan i poučan za diplomiranog studenta matematike i astronomije. Poznavajući samo par programskih jezika i nedovoljno upoznata sa tehničkim rečnikom računarstva i informatike, izazov je bio upoznati se sa osnovama računarske tehnologije, softvera i hardvera u cilju kvalitetnijeg savladavanja tehnike projektovanja paralelnih algoritama i implementacije na višeprocorskim računarima.

Ideju za pisanje paralelnog programa je sredinom 2009. godine predložio prof. dr Žarko Mijajlović. Sa tim ciljem sam došla u kontakt sa dr Predragom Jovanovićem koji je ideju o ubrzanju programa, koji već neko vreme koristi grupa istraživača na Astronomske opservatoriji u Beogradu, prihvatio. Od njega sam dobila sve potrebne osnovne informacije i sekvencijalni program koji je trebalo paralelizovati, odnosno, pripremiti za izvršavanje na klasteru Matematičkog instituta Srpske akademije nauka i umetnosti (SANU).

Tatjana S. Jakšić

*Ovaj master posvećujem svojoj porodici: ocu Slavku,  
majci Stanki  
i sestri bliznakinji Tijani.*

---

## Sadržaj

---

Predgovor . . . . .	ii
Posveta . . . . .	iii
<b>Uvod</b>	<b>1</b>
<b>1 Uvod u paralelne računare</b>	<b>3</b>
1.1 Primena . . . . .	3
1.1.1 Okruženje paralelnih računarskih sistema . . . . .	6
1.2 Istorija razvoja paralelnih računara . . . . .	6
1.2.1 Razvoj procesora . . . . .	6
1.2.2 Epohe računarstva . . . . .	7
1.2.3 Tradicionalni superračunari . . . . .	9
1.2.4 Razvoj modernih superračunara . . . . .	9
1.3 Programiranje paralelnih računara . . . . .	11
1.3.1 Razvoj softvera . . . . .	11
1.4 Arhitekture paralelnih računara . . . . .	13
1.4.1 Taksonomija arhitektura . . . . .	13
1.4.2 Komunikacija . . . . .	17
1.4.3 Računarske mreže . . . . .	18
1.4.4 Paralelne platforme . . . . .	20
1.5 Analiza Performansi . . . . .	22
1.5.1 Ubrzanje i efikasnost . . . . .	22
1.5.2 Amdalov zakon . . . . .	23
1.5.3 Karp-Flat metrika . . . . .	24
<b>2 Standardni interfejs za razmenu poruka, MPI</b>	<b>25</b>
2.1 Interfejs za prosleđivanje poruka - MPI . . . . .	25
2.1.1 MPI implementacije . . . . .	26
2.1.2 Projektovanje i implementacija MPI programa . . . . .	26

2.1.3	Primer MPI programa . . . . .	27
<b>3</b>	<b>Gravitaciona sočiva</b>	<b>31</b>
3.1	Uvod . . . . .	31
3.2	Gravitaciono savijanje svetlosti . . . . .	32
3.2.1	Njutnova i Ajnštajnova teorija . . . . .	32
3.2.2	Otkrića . . . . .	34
3.3	Teorija gravitacionih sočiva . . . . .	38
3.3.1	Karakteristike svetlosti . . . . .	38
3.3.2	Kosmološke osnove . . . . .	38
3.3.3	Ugao savijanja . . . . .	39
3.3.4	Jednačina sočiva . . . . .	41
3.4	Modeli sočiva . . . . .	52
3.4.1	Analitički modeli sočiva . . . . .	53
3.4.2	Tačkasti model sočiva . . . . .	53
<b>4</b>	<b>Gravitaciona mikrosočiva</b>	<b>58</b>
4.1	Model gravitacionih mikrosočiva . . . . .	60
4.1.1	Numerički metod . . . . .	61
<b>5</b>	<b>Rezultati paralelizacije i programske implementacije</b>	<b>63</b>
5.1	Arhitektura klastera MISANU . . . . .	63
5.1.1	Opis rada programa . . . . .	64
5.2	Izvor paralelizacije . . . . .	65
5.3	Rezultati . . . . .	68
5.3.1	Analiza vremena izvršavanja paralelnog programa . . . . .	68
5.3.2	Broj procesa. Karakteristike klastera . . . . .	74
5.4	Zaključak . . . . .	75
	<b>Reference</b>	<b>77</b>
<b>A</b>	<b>Dodatak</b>	<b>81</b>
A.1	Kratka istorija razvoja paralelnih računara . . . . .	81
A.2	Pajplajning . . . . .	82
A.3	Prosleđivanje poruka, MPI i PVM . . . . .	82
A.3.1	OpenMP . . . . .	83
A.4	MPI Tipovi podataka . . . . .	83
A.5	Klaster . . . . .	84
A.6	Transpjuteri . . . . .	85
A.6.1	Masivni višejezgarni sistemi . . . . .	86
<b>B</b>	<b>Dodatak</b>	<b>87</b>
B.1	Optička dužina puta . . . . .	87
B.2	Kritične krive i kaustici . . . . .	87
B.2.1	Dokaz za kritične krive . . . . .	87

---

B.2.2	Broj likova . . . . .	88
B.2.3	Vremensko zakašnjenje . . . . .	88

### Rezime

Cilj master rada *Paralelizacija i programska implementacija algoritma za izračunavanje mapa gravitacionih mikrosočiva* je paralelizacija i implementacija algoritma za generisanje mapa gravitacionih sočiva metodom gađanja zracima. Ove mape predstavljaju matrice pojačanja intenziteta svetlosti koja dolazi od nekog pozadinskog izvora. Mape se generišu na osnovu slučajnog rasporeda zvezda u ravni sočiva od kojih svaka pojačava svetlost emitovanu iz pozadinskog izvora i savija je pod određenim uglom.

Program, koji je napisan za sekvencijalno izvršavanje na jednoprocorskim računarima, za određene vrednosti ulaznih parametara uzima puno procesorskog vremena. Zahvaljujući međusobno nezavisnim računskim delovima programa, a pomoću biblioteke za prosleđivanje poruka MPI (Message Passing Interface), program je modifikovan za paralelno izvršavanje na klasteru Matematičkog instituta Srpske Akademije Nauka i Umetnosti (SANU) i tako značajno ubrzan proces generisanja mapa gravitacionih sočiva. Ovim je postignuto ubrzanje od 10 do 40 puta, u zavisnosti od veličine mape i vrednosti ostalih ulaznih parametara koji utiču na vreme izvršavanja sekvencijalnog programa.

### Struktura ove teze

Ovaj master rad sadrži dva dela: paralelno programiranje i teoriju gravitacionih sočiva.

Prvi deo rada pokriva razvoj arhitekture superračunara kao i okruženja za ovaj tip mašine. Čitaocu ovog rada želimo da predstavimo različite pristupe paralelnom programiranju. Pored velikog izbora programskih jezika, danas postoji i znatni izbor

u njihovoj nadogradnji u vidu biblioteka (MPI, OpenMP i druge) koje programerima omogućuju da razvijaju aplikacije koje se izvršavaju istovremeno na više procesora ili jezgara jednog višeprocorskog računara.

Drugi deo rada obuhvata uvod u teoriju gravitacionih sočiva u okviru kojeg će biti objašnjeno šta su to gravitaciona sočiva, koji su tipovi sočiva i izvođenje jednačine za ugao savijanja zraka svetlosti u blizini nekog gravitacionog sočiva.

Na kraju rada predstavljamo paralelizovani program kao i način paralelizacije, odnosno, funkcije iz MPI biblioteke koje smo koristili prilikom paralelizacije sekvencijalnog programa. Takođe dajemo testove performansi klastera (*eng.* benchmark test) na kojem smo izvršavali paralelni program.

U prvom poglavlju predstavljamo istoriju razvoja višeprocorskih računara i njihovu podelu, kao i razvoj softvera za paralelno programiranje.

U drugom poglavlju dajemo opis interfejsa za razmenu poruka među elementima višeprocorskih računara, koji smo iskoristili za paralelizaciju programa.

U trećem poglavlju predstavljamo teoriju gravitacionih sočiva.

U četvrtom poglavlju dajemo definiciju i numerički metod za rešavanje jednačine gravitacionih mikrosočiva.

U petom poglavlju se nalazi zaključak ovog rada u kome predstavljamo dobijene rezultate, odnosno ubrzanje paralelnog programa za izračunavanje mapa gravitacionih sočiva.



### 1.1 Primena

Prirodne nauke se zasnivaju na posmatranjima, teoriji i fizičkim eksperimentima. U okvirima klasičnih nauka teorija se gradi iz pretpostavke ili hipoteze koja se dokazuje empirijski (eksperimentalno) ili na osnovu prethodno utvrđenih zakona i teorema, zbog čega je modeliranje fizičkih procesa u bilo kom obliku suštinski deo svake klasične nauke.

Savremenu nauku karakteriše pristup modernim dostignućima modeliranja, gde su *numeričke simulacije* najzastupljenije. Razlog zbog kojeg numeričke simulacije imaju sve veći značaj je u uspešnoj imitaciji fizičkih eksperimenata bez kojih ranije nije bilo moguće ostvariti naučni doprinos. Možda najbitnija karakteristika primene numeričkih simulacija je rešavanje onih problema za koje je izrada opreme bila skupa ili nemoguća, ili bi dovela do narušavanja prirodne okoline i zdravlja ljudi.

Neke od naučnih oblasti u kojima numeričke simulacije imaju svakodnevnu primenu su:

- Relativistička fizika i statistička mehanika;
- Dinamička astronomija, astrofizika i kosmologija;
- Dinamika fluida i turbulencije;
- Dizajn materijala i superprovodljivost;
- Biologija, farmakologija, genetski inženjering;
- Medicina i modeliranje čovekovih organa i tkiva (npr. kostiju);
- Meteorologija i modeliranje prirodnih procesa u atmosferi.

Prilikom projektovanja i implementacije numeričkih simulacija sve više se susrećemo sa velikim brojem složenih problema čije rešavanje zahteva dobro opremljene računare. U nekim situacijama izvršavanje programa isuviše dugo traje, i zato se postavlja pitanje na koji način ubrzati njegov rad. Odgovor leži u konstruisanju paralelnih algoritama za izvršavanje na specijalnim (paralelnim) arhitekturama računarskih sistema.

Pre konstruisanja paralelnog algoritma potrebno je, pored poznavanja softvera, upoznati se i sa tipom arhitekture paralelnog računara na kojem će se dati algoritam implementirati. Pod softverskim znanjem podrazumeva se poznavanje nadogradnje postojećeg programskog jezika ili poznavanje paralelno-programskog jezika specijalno namenjenog za implementaciju paralelnih algoritama. Kao što smo naveli, za pisanje i izvršavanje paralelnog programa potrebno je imati i osnovnu informaciju o tipu mašine na kojoj će izvršavati program, jer nas ova informacija najviše upućuje na stil paralelizacije koji ćemo primeniti.

U ovom master radu se osvrćemo na različite tipove računarskih arhitektura i njihov razvoj kroz istoriju, kao i na razvoj softverskih paketa za paralelne računare. Njihov istorijski razvoj objašnjava zašto je došlo do razvoja različitih vrsta softvera i operativnih sistema za paralelne računarske sisteme, čime čitaocima želimo da ponudimo neka od rešenja za paralelizaciju programa.

Na kraju rada predstavljamo paralelizaciju jednog sekvencijalnog programa koji se koristi za izračunavanje mapa gravitacionih sočiva u cilju ispitivanja prednosti paralelnog izvršavanja na višeprosorskim računarima u odnosu na sekvencijalno izvršavanje istog koda na jednoprosorskim računarima.

## Šta je paralelni računar?

Paralelni računar je višeprosorski računar koji podržava istovremeno (*konkurentno*) izvršavanje dvaju ili više naredbi. Izdvajamo dve bitne kategorije paralelnih računara [28]: **višeračunarski sistemi** (eng. *multicomputers*) i **centralizovani višeprosorski računarski sistemi** (eng. *centralized multiprocessors*).

Višeračunarski sistem je sistem računara povezanih u računarsku mrežu putem koje procesori različitih računara međusobno komuniciraju slanjem i primanjem poruka.

Centralizovani višeprosor ili 'simetrični višeprosor'<sup>1</sup>, je visoko integrisani računarski sistem u kojem sve centralne procesorske jedinice<sup>2</sup> dele pristup globalnoj memoriji [28].

---

<sup>1</sup> eng. Symmetrical Multiprocessor, SMP

<sup>2</sup> eng. Central Processing Unit, CPU

## Konkurentnost

Za dva događaja kažemo da su *konkurentna* ako se realizuju u okviru istog vremenskog intervala. Za naredbe koje se izvršavaju na različitim računarima kažemo da se izvršavaju konkurentno ukoliko se njihova izvršavanja vremenski preklapaju. Konkurentnost ne mora uvek da se odnosi na potpuno vremensko preklapanje. Recimo da se prvi zadatak izvršava u okviru prvih 100 ms (milisekundi), nakon čega se u narednih 100 ms izvršava drugi zadatak dok prvi pauzira, i tako dalje; pri ovome svaki zadatak može uslovno da prelazi iz radnog stanja u stanje pauze. Kako period od jedne sekunde korisniku može delovati kao kratak period, opšti utisak je da se zadaci (naredbe) izvršavaju konkurentno.

U nastavku ćemo pod konkurentnošću podrazumevati istovremeno ili delimično preklapanje rada više procesora.

## Šta je paralelno programiranje?

Paralelno programiranje je programiranje u nekom programskom jeziku koje nam omogućuje da eksplicitno navedemo koji će se delovi programa izvršavati *konkurentno* (istovremeno). Konkurentnost je vrlo korisna apstrakcija za bolje razumevanje načina izvršavanja programa na jednoprocorskim i višeprocorskim računarima, budući da izraz 'paralelno', kao opšti termin, sada već ima istorijsku konotaciju. Najpopularniji naziv danas je *konkurentno programiranje* jer daje bolje objašnjenje o načinu izvršavanja instrukcija na procesoru.

Svaki konkurentni program sadrži više sekvencijalnih delova, te kako se sekvencijalni programi sastoje od više *niti* (eng. thread)<sup>3</sup>, nitna struktura se prenosi i na konkurentni program što obezbeđuje izvršavanje više procedura u isto vreme. Imajući u vidu ovakvu organizaciju jasno je da konkurentno programiranje omogućuje razne načine korišćenja računara. Može se reći da postoje tri načina kojima se realizuje konkurentno programiranje - višenitno, distribuirano i paralelno.

- Višenitni softverski sistemi su karakteristični za jednoprocorske računare i koriste se kod *multitaskinga*, odnosno, za simultano korišćenje više aplikacija na računaru (eng. multitasking).
- Distribuirani sistemi, odnosno sistemi više računara povezanih u lokalnu i globalnu računarsku mrežu, obezbeđuju rad sa udaljenim podacima (na primer, preko Web servisa). *Grid-computing* je poznati primer za ovakve sisteme.
- Paralelni računarski sistemi obuhvataju paralelne računare. Paralelni programi omogućuju podelu podataka ili zadataka između više komponenti paralelnog računara.

---

<sup>3</sup> Nit predstavlja jedinicu izvršnog toka unutar procesa.

U nastavku ćemo izraz 'paralelno programiranje' koristiti za rešavanje problema pomoću programa namenjenih isključivo za izvršavanje na paralelnim računarima.

### 1.1.1 Okruženje paralelnih računarskih sistema

Paralelni računarski sistem je računarski sistem koji sadrži dva ili više procesora, okruženje u kojem se mogu projektovati paralelni programi i paralelni program koji će se izvršavati.

Za adekvatno okruženje je potrebno imati:

- Operativni sistem;
- Sisteme za pravilno uspostavljanje komunikacije među procesima pomoću:
  - ◇ Razmene poruka:
    - MPI (Message Passing Interface)
    - PVM (Parallel Virtual Machine)
  - ◇ Deljene memorije
    - OpenMP
    - Shmem

Programiranje paralelnih računara možemo podeliti u sledeće faze razvoja:

1. Faza identifikacije delova sekvencijalnog algoritma ili podataka kao nezavisnih procesa<sup>4</sup>.
2. Faza distribucije procesa koji će se u isto vreme izvršavati na različitim procesorima.
3. Faza koordinacije rada i komunikacije između procesa.

Pre projektovanja paralelnog koda za konkretan paralelni računarski sistem moramo razmotriti njegov *tip arhitekture* i *način komunikacije*. Ova dva uslova predstavljaju jedan od ključnih parametara pomoću kojih razvrstavamo veliki broj današnjih paralelnih računarskih sistema.

## 1.2 Istorija razvoja paralelnih računara

### 1.2.1 Razvoj procesora

Na razvoj paralelnih računarskih sistema najviše je uticao razvoj jednoprocesorskih (klasičnih) računarskih sistema. Klasični procesori izvršavaju instrukcije jednu za drugom, odnosno, nova instrukcija se ne izvršava dok prethodna ne završi

---

<sup>4</sup> Ovde pod *procesom* podrazumevamo niz instrukcija (tok instrukcija) koji manipulišu nizom operanada (tokom podataka).

sa svojim radom. Ovo je često (ako ne i uvek) dovodilo do neiskorišćenosti resursa kojima su klasični procesori raspolagali, zbog čega se unapređivanje istih orijentisalo na pojačanu eksploataciju komponenti procesora. Nasuprot sekvencijalnoj obradi, paralelna obrada zadataka istovremeno koristi više elemenata računarskog sistema putem podele na delove koji su međusobno nezavisni.

Pokušaji da se postignu što bolje performanse podstakli su razvoj raznih dizajnerskih metodologija, zbog čega su se novi procesori ponašali sve više paralelno. Povećanje radnog takta<sup>5</sup> je dugo vremena bila dominantna karakteristika jednoprosorskih mašina (1980 - 2004). Svakih nekoliko godina pojavljivao se novi dizajn koji bi se u narednim godinama poboljšavao sitnim prepravkama, optimizacijama i povećanjem radnog takta (šezdesetčetvorobitni procesori odavno su postali uobičajeni na tržištu). Potrošnja energije jednog čipa koja je opisana jednačinom

$$P = C \times V^2 \times F \quad (1.1)$$

gde je  $P$  snaga,  $C$  kapacitet (proporcionalan broju tranzistora),  $V$  napon a  $F$  radna frekvencija (broj ciklusa u sekundi) se, ukoliko su ostale vrednosti konstantne, povećava sa povećanjem radne frekvencije procesora zbog čega je ubrzanje procesora na ovaj način postalo izuzetno skupo i teško. Ovo je doprinelo završetku razvoja tehnologije čipa baziranog na povećanju radnog takta procesora.

Interesantno je da se razvoj procesorskih čipova u periodu od 70-tih godina naovamo može opisati *Murovim zakonom*<sup>6</sup> koji tvrdi da se broj tranzistora po jedinici površine duplira na svakih 18 do 24 meseci. Bez obzira na završetak razvoja jednoprosorskih čipova Murov zakon je još uvek na snazi kada je brzina procesora u pitanju i to isključivo zahvaljujući razvoju paralelnih računarskih sistema. Umesto povećanja procesorske snage rastom radnog takta (što uzrokuje visoke temperature prilikom rada računara) danas proizvođači čipova brzinu procesora uvećavaju dodavanjem CPU-ova, ili "jezgara", u sklop mikroprocesorskog paketa.

## 1.2.2 Epohe računarstva

Može se reći da razvoj računarstva karakterišu dve epohe:

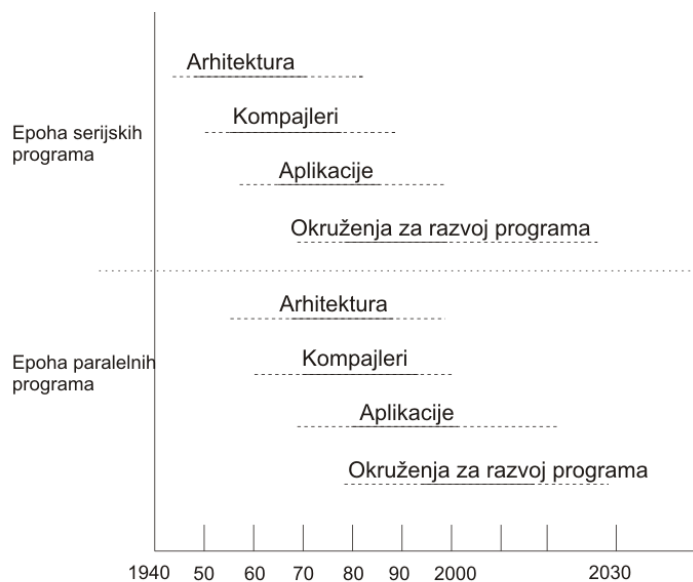
1. Epoha sekvencijalnog i;
2. Epoha paralelnog pristupa.

Na slici 1.1 vidimo da su epohe sekvencijalnog i paralelnog računarstva započete ne tako davno u istoriji i u međusobnom rasponu od samo par godina, te da se sa

---

<sup>5</sup> Vreme izvršavanja programa se računa kao proizvod broja instrukcija i srednjeg vremena potrebnog za izvršenje jedne instrukcije. Sa povećanjem radne frekvencije vremena smanjivalo se srednje vreme potrebno za izvršavanje jedne instrukcije.

<sup>6</sup> Gordon Moore



Slika 1.1: Epohe razvoja računarstva

paralelnim računarstvom krenulo daleko ranije nego što je to možda utisak koji nam ostavlja dostupnost paralelnim računarima danas.

Pre projektovanja prvih paralelnih programa proizvođači su uveliko radili na razvoju arhitekture računara. Tačnije, prvih par decenija su bile posvećene istraživanju i razvoju (kao što je razvoj kompajlera za paralelnu vrstu mašina), a potom se započelo sa pisanjem aplikacija radi kontrolisanja rada paralelnih računarskih sistema.

### 1.2.3 Tradicionalni superračunari

Superračunar je računar posebne namene koji predstavlja najbolju vrstu računara svog doba kada je reč o brzini njegovog rada. Termin je dosta fluidan jer ono što su danas superračunari sutra postaju "obični" (PC, stoni) računari.

Superračunari su ranije bili jednoprocorski, konstruisani za brzo izvršavanje operacija. Zahvaljujući razvoju računara široke potrošnje procesori visoke snage su počeli da se koriste svakodnevno dok su bolje verzije procesora postale ograničene skupom izradom, otežanim hlađenjem i fizičkim limitima prilikom proizvodnje komponenti. Danas pod superračunarima podrazumevamo paralelne računare, odnosno, računare konstruisane za paralelno izvođenje programa na više procesora.

Prednosti višeprocorskih nad jednoprocorskim superračunarima su:

- a) Rešavanje problema koji se dugo izvršavaju na jednom procesoru;
- b) Korišćenje gotovih komponenti pri sastavljanju;

- c) Mogućnost stalnog proširenja novim komponentama;
- d) Daleko jeftinija izrada u odnosu na jednoprocesorske tipove superračunara.

Nedostaci današnjih superračunara su:

- a) Neophodnost upoznavanja sa novim tehnologijama, što može da oduzme dosta vremena;
- b) Potreba za dodatnim modifikacijama standardnih (serijskih) programa.
- c) Otežano upravljanje izlazno/ulaznim uređajima, za razliku od jednoprocesorskih superračunara.

### 1.2.4 Razvoj modernih superračunara

Paralelni računari su privukli širok krug korisnika zahvaljujući razvoju čipova sa visokim stepenom integrisanosti<sup>7</sup> pomoću kojih je, krajem 70-tih godina, omogućeno pravljenje integriranih kola kombinovanjem hiljada tranzistorskih jedinica po jednom čipu i započinjanjem proizvodnje pouzdanih paralelnih računarskih sistema.

Godine 1981. grupa sa Caltech Univerziteta u SAD je pokrenula projekat konstruisanja paralelnog računara "Cosmic Cube" sastavljenog od 64 Intelovih 8086 mikroprocesora koji je svojim radom ukazao na potencijal programiranja paralelnih računara na bazi mikroprocesora. Brzina izvršavanja programa je iznosila oko 5 miliona operacija sa pokretnim zarezom u sekundi (5 MFlops), što je bilo 5 puta brže od tadašnjeg standardnog miniračunara. Cosmic Cube predstavlja prvi pouzdan paralelni računar napravljen na bazi svakodnevnih (jeftinih) računara.

Naredni period karakteriše izuzetna aktivnost raznih kompanija koja su na tržište iznele sopstvene verzije paralelnih računara. Firme kao što su Meiko, nCube i Parsytec su bile vodeće dok su ostale firme (IBM, NEC, Sun Microsystems) čekale povoljniji trenutak.

Do sredine 90-tih godina veliki broj firmi nije nastavio sa daljim implementacijama i projektovanjima paralelnih računarskih sistema. Bez obzira na slabu potražnju, velike kompanije kao što su Hewlett-Packard, IBM, Silicon Graphics i Sun Microsystems su na tržište iznele višeprocesorske računare sa cenama u rasponu od nekoliko stotina hiljada do nekoliko miliona dolara. U poređenju sa ličnim računarima (PC - Personal Computers), cena po CPU u komercijalnim višeprocesorskim računarima je bila veća zbog hardvera koji je podržavao ili deljenu memoriju ili dobru komunikaciju.

Sistemske alati za programiranje kod komercijalnih višeprocesorskih računara su bili primitivnije izrade, zbog čega su ih istraživači sami konstruisali koristeći programske jezike C i FORTRAN u kombinaciji sa standardnim bibliotekama za prosleđivanje poruka kao što su PVM i MPI.

---

<sup>7</sup> eng. Very Large Scale Intergration, VLSI

Zahvaljujući popularnosti PC računara formirala se osnova za novu pojavu paralelnih arhitektura na tržištu. Godine 1994. u NASA-inom centru je konstruisan paralelni računar isključivo na bazi svakodnevnog hardvera i slobodnog softvera ([28]). Ovaj sistem od 16 Intelovih DX4 procesora povezanih 10Mbit/sec Ethernet<sup>8</sup> vezom su nazvali **Beowulf klaster**. Ovaj klaster je zasnovan na Linux operativnom sistemu, sa GNU kompajlerima, a paralelno programiranje je podržano pomoću biblioteke za prosleđivanje poruka - MPI. Ubrzo je Beowulf prihvaćen kao standardna paradigma gradnje paralelnih računara.

### Zašto paralelno programiranje?

Mnoge aplikacije danas zahtevaju više procesorske snage nego što omogućuju tradicionalni računari. Takođe, programiranje paralelnih računara daje bolje rešenje kada su u pitanju odnos cene i efikasnosti.

Neki od faktora koji utiču na dalji razvoj višeprocorskih računarskih sistema su:

- Potrebe za bržim računanjem u oblasti nauke i finansija.
- Hardverska poboljšanja za implementaciju modela pajplajninga<sup>9</sup> (detaljnije u dodatku A.2), superskalara i drugih modela računarskih sistema zahtevaju razvoj sofisticirane računarske tehnologije. Razvoj prevodioca (kompajlera) za takve sisteme je izuzetno težak zadatak i zahteva konstantan rad i poboljšanja.

## 1.3 Programiranje paralelnih računara

Programiranje paralelnih računara je složenije od programiranja jednoprocesorskih računara. Takođe, analiza efikasnosti algoritama i vremena izvršavanja programa na višeprocorskim računarima je složenija u odnosu na vremensku analizu sekvencijalnih algoritama.

U programiranju paralelnih računara cilj je minimizacija vremena izvršavanja, a preduslov za to je maksimalna iskorišćenost procesora. Ove vrednosti zavise direktno od prirode problema, odnosno od toga da li se i kako problem može razložiti na potprobleme koji ne zavise jedni od drugih, a koji bi se mogli istovremeno izvršavati na različitim procesorima.

### 1.3.1 Razvoj softvera

Godine 1988. Mek Grou (McGraw) i Akselrod (Axelrod) su identifikovali četiri različita načina za razvoj aplikativnog softvera na paralelnim računarima ([28]):

---

<sup>8</sup> *Eternet* je tip mrežnog protokola u računarskim sistemima.

<sup>9</sup> *Pajplajning* najbolje ilustruje pokretna traka



1. Proširenje postojećeg kompajlera radi prevođenja sekvencijalnih u paralelne programe;
2. Proširenje postojećeg jezika novim operacijama koje bi omogućile korisnicima da eksplicitno navedu paralelne delove programa;
3. Dodavanje paralelno-programerskog nivoa;
4. Konstruisanje potpuno novog paralelnog jezika i kompajlerskog sistema.

Većina pobrojanih predloga je realizovana, a neki od njih se danas više ističu u odnosu na ostale. Da bismo videli koje su njihove prednosti opisaćemo ukratko sve navedene predloge.

### Proširenje kompajlera

Jedan od načina da se reše problemi kod programiranja paralelnih računara je razvijanje paralelnih kompajlera koji sami detektuju i koriste paralelizam u programima napisanim sekvencijalnim jezikom.

Razvoj kompajlera za paralelne programe traje već više od dvadeset godina. Neke kompanije, kao što je Parallel Software Products su ponudile kompajlere koji kodove, pisane pomoću FORTRAN 77, prevode za izvršavanje na paralelnim računarima sa komunikacijom baziranom na prosleđivanju poruka ili deljenoj memoriji.

Ovakav pristup ima svoje nedostatke, naročito u situacijama u kojima kompajler ne prepoznaje one delove programa koji bi bili pogodni za paralelno izvršavanje. Jedno od rešenja je uvođenje predprocesorskih direktiva kojima se kompajler usmerava.

### Proširenje sekvencijalnih programskih jezika

Daleko konzervativniji pristup u razvoju paralelnih programskih okruženja je proširenje sekvencijalnih programskih jezika funkcijama koje omogućuju programu da pokrene ili prekine izvršenje paralelnih procesâ, sinhronizuje ih, i kontroliše njihovu komunikaciju.

Ovakav pristup u razvoju softvera za paralelne računare je najlakši, najbrži, najjeftiniji i najpopularniji jer se od programera zahteva samo razvoj biblioteka potprograma<sup>10</sup>. Kako kompajler ne učestvuje u proizvodnji paralelnog koda on nije u stanju da uvek prepozna greške, što predstavlja najveći nedostatak kod ovakvog modela.

### Dodavanje novog nivoa

Za strukturu paralelnog programa možemo reći da se sastoji iz dva nivoa. Na nižem nivou program sadrži jezgro računa koje se projektuje sekvencijalnim kodom.

---

<sup>10</sup> *eng.* Subroutine library

Viši nivo kontroliše stvaranje i sinhronizaciju pojedinačnih procesa i deljenje podataka između njih, te ove akcije možemo opisati korišćenjem paralelnog jezika. U tom slučaju je kompajler odgovoran za prevođenje struktura u jedan kôd pogodan za izvršavanje na paralelnom računaru. Ovakav pristup zahteva upoznavanje sa novim paralelnim računarskim sistemima, čime se objašnjava zbog čega dodavanje novog nivoa nije postalo popularno i nije doživelo svoju konačnu realizaciju.

### Konstrukcija jezika za paralelno programiranje

Četvrti pristup je da se omogući eksplicitno navođenje paralelnih operacija. Jedan od načina da se podrži eksplicitno paralelno programiranje je da se paralelni jezik razvije od početka. Poznat primer je programski jezik **Occam**. Sa različitom sintaksom u odnosu na tradicionalne imperativne jezike **Occam** podržava i sekvencijalno izvršavanje procesa i njihovu automatizovanu komunikaciju i sinhronizaciju.

Drugi način da se podrži eksplicitno paralelno programiranje je da se u postojeći sekvencijalni jezik dodaju konstrukcije koje omogućavaju paralelizaciju ili paralelno izvršavanje (Fortran 90, High Performance Fortran, C\*).

### Trenutni status

Najpopularniji pristup paralelnom programiranju danas je pristup nadogradnje postojećih sekvencijalnih jezika konstrukcijama nižeg nivoa preko poziva odgovarajućih funkcija ili pomoću predprocesorskih direktiva. Paralelno programiranje u C-u ili Fortranu sa MPI-om i/ili OpenMP-om je poznat primer ovakvog pristupa. Paralelno programiranje u nekom programskom jeziku nižeg nivoa (*eng.* low-level) može da proizvede programe visoke efikasnosti i lake prenosivosti u širokom spektru paralelnih sistema. Nedostatak primene ovakvog modela se javlja u vidu otežanog konstruisanja koda i prilikom otklanjanja greške (*eng.* debugging), za razliku od viših programskih jezika kao što su Fortran, C ili C++.

## 1.4 Arhitekture paralelnih računara

### 1.4.1 Taksonomija arhitektura

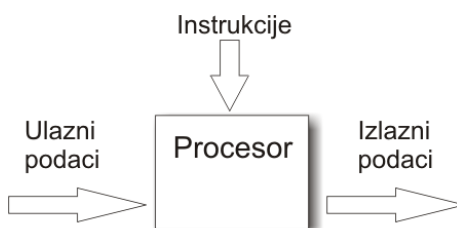
Zajednička karakteristika svih paralelnih računarskih sistema je upotreba više od jednog procesora. Međutim, izvršiti klasifikaciju nije jednostavno s obzirom da se pri izradi paralelnih računara mogu koristiti gotove komponente koje nisu bile specijalno pravljene za tu namenu (npr. **klasteri**, detaljnije u A.5). U ovoj sekciji pravimo pregled osnovnih tipova računarskih sistema.

Jedna od najistaknutijih podela je *Flinova podela*<sup>11</sup> na osnovu koje se svi računarski sistemi dele prema tipu arhitekture. Ova klasifikacija se bazira na paralelizmu u podacima i instrukcijama, a fokus je stavljen na rad hardvera koji izvršava ove instrukcije tako što je u stanju da podrži jednu ili više njih koje manipulišu sa jednim ili više tokova podataka. Flinovom podelom se uvode četiri kategorije računarskih arhitektura:

1. SISD - Jednostruki tok instrukcija i jednostruki tok podataka (*eng.* Single Instruction Single Data)
2. SIMD - Jednostruki tok instrukcija i višestruki tok podataka (*eng.* Single Instruction Multiple Data)
3. MISD - Višestruki tok instrukcija i jednostruki tok podataka (*eng.* Multiple Instruction Single Data)
4. MIMD - Višestruki tok instrukcija i višestruki tok podataka (*eng.* Multiple Instruction Multiple Data)

U tekstu koji sledi biće dat detaljniji opis svake od ovih arhitektura.

### SISD model računarskih arhitektura



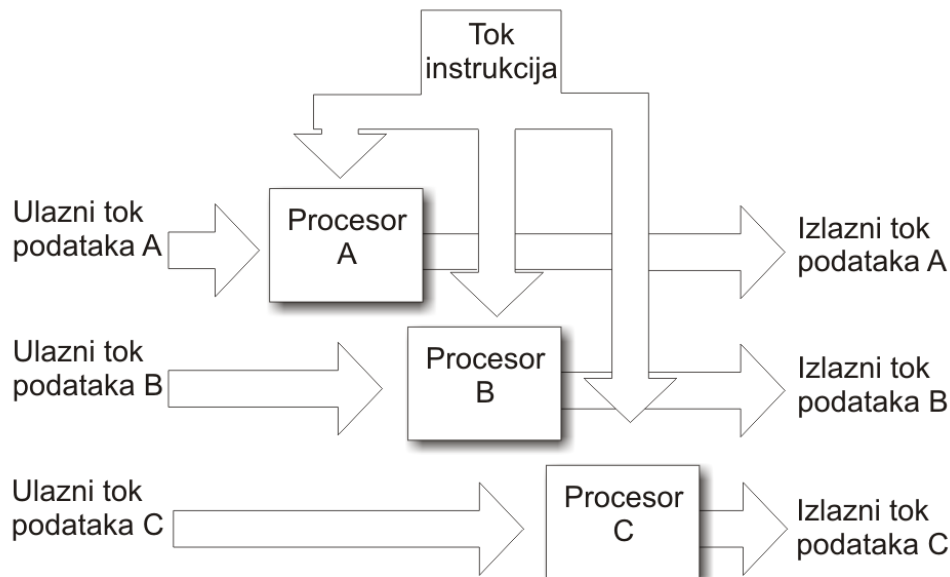
Slika 1.2: Šematski prikaz SISD modela računarskih arhitektura.

SISD model procesora je sposoban da izvršava jedan tok instrukcija i operiše nad jednim tokom podataka, te u ovu grupu procesora spadaju jednoprocesorski računarski sistemi. U SISD procesorima instrukcije se izvršavaju jedna za drugom (sekvencijalno), zbog čega ih još nazivamo i *sekvencijalnim računarima*. Najveći broj današnjih računara sadrži SISD model procesora. Instrukcije i podaci se smeštaju u lokalnu memoriju (videti sliku 1.2), te je brzina jednog procesorskog elementa u SISD modelu ograničena brzinom kojom računar prebacuje podatke.

<sup>11</sup> Podelu je godine 1966. predložio Majkl Flin (*eng.* Michael J. Flynn).

## SIMD model računarskih arhitektura

SIMD model procesora se odnosi na računare u kojima svaki od  $N$  identičnih procesora sa zajedničkom kontrolnom jedinicom izvršava isti program nad svojim lokalnim podacima (videti sliku 1.3). Izvršavanje je sinhrono, odnosno, svaki procesor u isto vreme izvršava istu instrukciju ali za drugačiji tok podataka. Poznati primer SIMD procesora je *vektorski* procesor<sup>12</sup>.



Slika 1.3: Šematski prikaz SIMD modela računarskih arhitektura

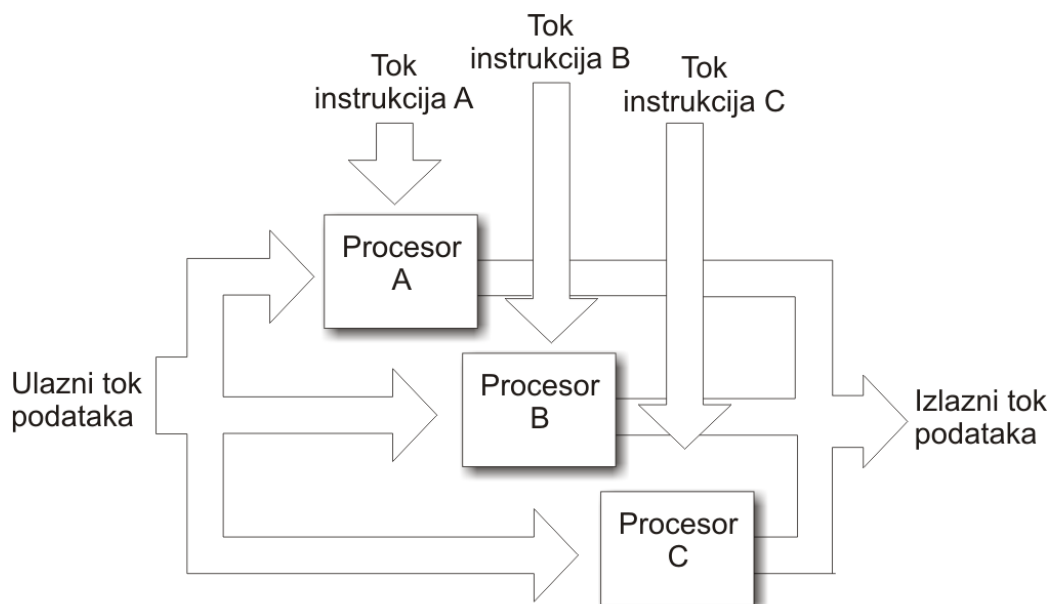
Mašine bazirane na SIMD modelu su korisne za rad u naučnim oblastima u kojima se često koriste operacije nad vektorima i matricama. Primer takve konstrukcije je *niz procesora*<sup>13</sup> sa jednom procesorskom kontrolnom jedinicom i nizom podređenih procesora.

Najstarija paralelna mašina ILIAC je imala SIMD arhitekturu i sastojala se od 32 procesorska elementa povezanih u prsten, od kojih se svaki element sastojao od jednog procesora i lokalne memorije. Iako se SIMD koristio uglavnom na vektorskim superračunarima, poput onih koje je 70-tih godina popularisao Cray, potreba za obradom multimedijalnih podataka je dovela do dodavanja SIMD instrukcija u arhitekturu procesora opšte namene. Ovaj trend je izrodio tehnologije kao što su: PowerPC-jev AltiVec; Intel-ovi MMX, SSEx; AMD-ov 3Dnow, i mnoge druge.

<sup>12</sup> Vektorski procesor čini niz procesora, u kojem se ista operacija u istom trenutku izvršava nad različitim podacima svakog procesora

<sup>13</sup> *eng.* processor array

## MISD model računarskih arhitektura



Slika 1.4: Šematski prikaz modela MISD računarskih arhitektura

MISD je model višeprocorskog sistema gde  $n$  procesora, svaki sa po jednom kontrolnom jedinicom, dele isti tok podataka. Svi procesori, u svakom koraku, obrađuju istovremeno jedan podatak prihvaćen iz zajedničke memorije, prema instrukciji koju svaki procesor prima sa svoje upravljačke jedinice (videti sliku 1.4). Primena ovakvog tipa računara je pogodna za:

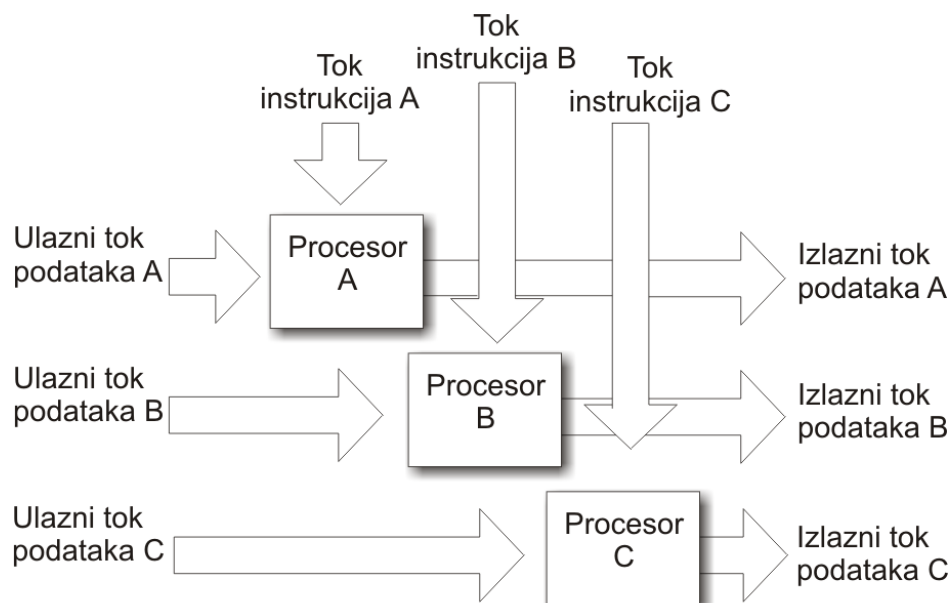
1. Primenu filtra za izdvajanje signala iz ulaznog niza podataka;
2. Izvršavanje različitih kriptografskih algoritama za razbijanje šifri.

## MIMD model računarskih arhitektura

MIMD računarski sistemi su najstroženiji u kome svaki procesor poseduje sopstvenu lokalnu memoriju. MIMD višeprocorske mašine su sposobne da izvrše više instrukcija nad različitim skupom podataka. Način na koji se instrukcije izvršavaju je asinhron, što znači da se u istom trenutku mogu izvršavati različite instrukcije nad različitim podacima (videti sliku 1.5).

U kategoriju MIMD modela spada najveći broj današnjih superračunara. Prednosti MIMD modela u odnosu na ostale modele iz Flinove podele su:

- MIMD procesori mogu da izvršavaju više različitih poslova za isto vreme;
- Svaki MIMD procesor je u stanju da bilo koju operaciju izvršava nezavisno od rada drugih procesora.



Slika 1.5: Šematski prikaz modela MIMD računarskih arhitektura

Nedostaci ovog modela su sledeći:

1. Projektovanje programa može biti teško;
2. Degradacija performansi koja se javlja usled raspoređivanja (balansiranja) opterećenja (*eng.* load balancing overhead), što je posledica asinhronog izvršavanja instrukcija. Raspoređivanje opterećenja (*eng.* load balancing) su tehnike koje služe za raspoređivanje različitih zadataka među procesorima paralelnog računara kako bi se izbeglo da jedan procesor ne radi dok se kod drugih zadaci gomilaju. Neki sistemi omogućavaju migraciju zadataka, odnosno prenošenje zadatka sa jednog na drugi proces i to se postiže tako što procesori koji imaju nagomilane zadatke pošalju zadatak slobodnom procesoru. Tokom raspoređivanja opterećenja najbitnije je da se ukupna brzina izvršavanja ne smanjuje.

Postoje dva tipa MIMD arhitekture, sa deljenom i distribuiranom memorijom, odnosno *čvrsto vezani* procesorski sistem i *slabo vezani* procesorski sistem, o čemu će biti više reči kasnije. MIMD računari predstavljaju izuzetno široku grupu računara koja u svom sastavu može da sadrži elemente računara iz različitih kategorija Flynnove podele. Tako, na primer, poznati MIMD računari sa SISD komponentama su: Sequent, nCUBE, Intel iPSC/2, IBM RS6000 klaster i dr.

Poznati MIMD računari sastavljeni od paralelnih SIMD mašina su: Cray C 90, Fujitsu VP 2000, Intel Paragon, IBM SP1 i dr.

## 1.4.2 Komunikacija

U praksi se Flinova podela ne koristi s obzirom da su moderni računarski sistemi mešavina različitih modela Flinove klasifikacije<sup>14</sup>.

Postoje i druge podele paralelnih računarskih sistema; prema arhitekturi i skupu instrukcija ili po širini adresne magistrale. Izdvajamo podelu prema načinu komunikacije između procesa jer dobro oslikava različite realizacije u praksi ([12]):

1. Višeprocorski sistemi sa komunikacijom preko **deljene memorije** (*eng.* shared memory multiprocessors).
2. Višeprocorski sistemi sa komunikacijom **prosleđivanjem poruka** (*eng.* message passing multiprocessors).

### Komunikacija preko deljene memorije

U ovom modelu svi procesi imaju jedinstven memorijski prostor zajednički za sve procesore. Kako operativni sistem mora sam da vodi računa o paralelnom čitanju i upisu, za posledicu imamo olakšano korišćenje i programiranje ovakvih računara. Operativni sistem teško može da obezbedi efikasno paralelno čitanje i upis podataka, pogotovo kod paralelnih računara sa velikim brojem procesora.

OpenMP standard prilično dobro rešava datu anomaliju putem *paralelizma u više nivoa*<sup>15</sup>. S obzirom da OpenMP relativno jednostavno podržava dodavanje paralelnih konstrukcija u postojeći sekvencijalni kod, on je postao najčeće korišćen model kod višeprocorskih sistema sa komunikacijom preko deljene memorije.

### Komunikacije prosleđivanjem poruka

Sa pojavom paralelnih računara sa više stotina ili hiljada procesora, odnosno korišćenjem mreže radnih stanica i Interneta, postali su popularni paralelni računari gde se međuprocorska komunikacija ostvaruje *prosleđivanjem poruka*<sup>16</sup>. Projektovanje i implementiranje takvih sistema je relativno lako, ali je na njima teže programirati, jer program (a ne operativni sistem!) vodi računa o sinhronizaciji procesa i razmeni podataka između njih.

Međuprocorska komunikacija je kod ovih sistema jedan od presudnih faktora koji utiče na ukupno vreme izvršavanja. Vreme potrebno za prenos određenog podatka

---

<sup>14</sup> Na primer, dual Pentium 4 Xeon je u osnovi x86 SISD arhitektura sa SIMD proširenjima koji koristi simetrično multiprocorsanje na dva odvojena mikroprocesora, a svaki od njih sadrži jezgro sa pajplajn arhitekturom (videti u dodatku) koja se može shvatiti kao MISD (jer svaki korak u pajplajnu operiše nad istim tokom podataka).

<sup>15</sup> *eng.* Multilevel parallelism; detaljnije u dodatku A.2.

<sup>16</sup> *eng.* message passing.

ili grupe podataka, između različitih procesora je ponekad nekoliko puta veće od vremena potrebnog za aritmetičku operaciju nad njima ([21]). Obično veća iskorišćenost procesora zahteva i veću međuprocesorsku komunikaciju, i obrnuto, te se mora naći kompromis koji obezbeđuje minimalno vreme izvršavanja.

U početku je svaki proizvođač paralelnih računara, u cilju razvoja i implementacije paralelnih aplikacija, konstruisao sopstvene programske sisteme (Cray Research, IBM, Meiko, Ncube i transpjuteri). Ovakvi sistemi su se međusobno dosta razlikovali što je uticalo na prenosivost aplikacija između njih. Ovo je uslovalo pokušaje uvođenja standarda (iz čega su nastali sistemi kao što su CHIMP, p4, PVM, Zipcode). Međutim, svaki od ovih sistema je posedovao određene nedostatke i umesto nadogradnje nekog od njih pristupilo se konstruisanju novog sistema.

### 1.4.3 Računarske mreže

Paralelni računarski sistemi prema gruboj podeli mogu komunicirati preko deljene memorije i prosleđivanjem poruka. Ono što može da nas interesuje su implementacije ovih komunikacija čime dolazimo do načina povezivanja procesora prilikom konstrukcije superračunara.

Izdvajamo dva tipa medijuma pomoću kojih se ostvaruje komunikacija u paralelnom računarskom sistemu ([28]):

1. deljeni medijumi;
2. medijumi sa skretnicama<sup>17</sup>.

Deljeni medijum dozvoljava slanje jedne poruke u jednom trenutku, što uslovljava konstantno "prisluškivanje" procesora u cilju primanja poruke. Ethernet predstavlja najbolji primer ovakvog medijuma. Ukoliko dva procesora pošalju poruku u isto vreme, poruka se gubi zbog čega se ponavlja postupak slanja.

Medijumi sa skretnicama se zasnivaju na deljenim medijumima tako što pored Ethernet veze sadrže skretnice koje podržavaju *pojedinačne međuprocesorske komunikacije*<sup>18</sup>, a svaki procesor ima sopstvenu vezu do skretnice. Skretnice karakterišu dve velike prednosti: podržavaju istovremenu razmenu poruka između više parova procesora i dozvoljavaju proširivanje mreže ugradnjom novih skretnica.

### Topologija računarskih mreža

Najzastupljenije topologije računarskih mreža u današnjim komercijalnim paralelnim računarskim sistemima su:

- 2-D Mreža (2-D Mesh),

---

<sup>17</sup> *eng. switched* medijumi

<sup>18</sup> *eng. point-to-point communications*



- Hiperdrvo,
- Leptir (Butterfly),
- Hiperkocka.

Mreže predstavljamo grafovima u kojima su procesori definisani čvorovima, a putevi komunikacije skretnicama i granama. Svaki procesorski čvor je povezan sa jednom skretnicom, a skretnice povezuju procesore i/ili druge skretnice.

U navedenim topologijama razlikujemo *direktne* i *indirektne* topologije. U direktnim topologijama odnos skretnica čvorova i procesora je 1 : 1, odnosno, svaka skretnica je povezana sa jednim procesorskim čvorom i sa jednom ili više drugih skretnica. U indirektnoj topologiji, odnos broja skretnica prema broju procesorskih čvorova je veći od 1 : 1.

Od tipa topologije će zavistiti efikasnost paralelnog računarskog sistema, a za opis efikasne konstrukcije koristimo sledeće parametre:

- *Prečnik mreže*: najveća udaljenost između dva čvora skretnice;
- *Broj grana po skretnici*: u najboljem slučaju je ovaj broj konstantan;
- *Širina podele mreže* (*eng.* bisection width): najmanji broj ivica između čvorova skretnica bez kojih se mreža deli na dve mreže unutar jedne;
- *Konstantna dužina grane grafa*.

Svaki od parametara opisuje neku od karakteristika mreža kod paralelnih računarskih sistema na osnovu kojih se procenjuje dalja konstrukcija fizičkog sistema. Ovde nećemo ulaziti u detalje, nego ćemo samo navesti da klaster "Matematički Institut - SANU" čini rekonfigurabilna mreža u kojoj su svih 16 čvorova međusobno povezani preko skretnica, što korisnicima klastera MISANU omogućuje uspostavljanje raznih topologija računarskih mreža.

#### 1.4.4 Paralelne platforme

Računarske platforme za razvoj, testiranje i izvršavanje programa se mogu podeliti u nekoliko globalnih kategorija ([12]):

1. Simulatori i emulatori višeprocorskih računara na jednoprocorskim računarima;
2. Višeprocorski sistemi nižih performansi;
3. Mreža radnih stanica povezanih u paralelni računar;
4. Superračunari i višeprocorski sistemi visokih performansi.

## Simulatori paralelnih računara

Pomoću simulatora ili emulatora paralelnih računara olakšava se upoznavanje sa paralelizacijom i paralelnim programiranjem. Njihova namena se može okarakterisati kao:

1. Modeliranje nekog realnog višeprocorskog sistema radi detaljnijeg upoznavanja sa datim sistemom i eventualno razvijanje konkretnih aplikacija. Sve pripreme radnje u razvoju date aplikacije se odrađuju na simulatoru, a konačno izvršavanje aplikacije se realizuje na konkretnom paralelnom računaru.
2. Modeliranje nekog imaginarnog višeprocorskog sistema radi olakšanog učenja i upoznavanja sa paralelnim konstrukcijama. U ovim slučajevima kompatibilnost sa ostalim sistemima i performanse datog simulatora nisu od velikog značaja.

Simulatori ili emulatori su uglavnom namenjeni za učenje i obično nisu pogodni za širu primenu. Iako je teorijski moguće simulirati sve elemente paralelnog računara, u praksi rezultati u velikoj meri odstupaju od očekivanih. Simulacija na jednoprocorskom računaru obično ne može predstaviti na pravi način neke vrlo važne fizičke karakteristike paralelnog računara što se ogleda u: relativno sporoj međuprocorskoj komunikaciji, problemima sa paralelnim čitanjem/pisanjem u memoriji, različitim osobinama pojedinačnih procesora unutar paralelnog računara i nekim problemima vezanim za ostale osobine realnog paralelnog računara. Rezultati dobijeni na simulatoru mogu biti potpuno nereprezentativni, pa čak i netačni, i obično u velikom neskladu sa rezultatima iste aplikacije na realnom paralelnom računaru. Pošto su u početku višeprocorski računari bili uglavnom vrlo skupi i nepristupačni, ovakav pristup je često korišćen i pored pomenutih nedostataka.

## Višeprocorski sistemi nižih performansi

Početkom 80-tih godina su se prvi put pojavili šire dostupni višeprocorski sistemi relativno prihvatljive cene i performansi (transpjuteri, ICUBE, itd). Transpjuterski sistemi su bili najpoznatiji predstavnici ove klase paralelnih računara, zasnovanih na osnovnim procesorima - transpjuterima (videti u dodatku A.6), uz međuprocorsku komunikaciju prosleđivanjem poruka. Ovi sistemi su realizovani kao ploče koje su se ugrađivale u PC računare ili Sun radne stanice, a koji obezbeđuju ulazno/izlazne operacije korišćenjem postojećih operativnih sistema računara domaćina (*eng.* host computers). Paralelne konstrukcije su ugrađivane u postojeće programske jezike (paralelni C) ili su razvijani potpuno novi programski jezici prilagođeni paralelnom izvršavanju (Occam).

Bez obzira na tehničku ograničenost, često kvarenje i mali broj razvijenih programskih alata, transpjuterski sistemi su u vreme pojavljivanja doprineli širokoj popularizaciji paralelnih računara i upoznavanju većeg broja ljudi sa paralelnim konstrukcijama na realnom višeprocorskom računaru.

### Mreža radnih stanica

Radne stanice i stoni računari predstavljaju najzastupljenije platforme za razvoj, testiranje i izvršavanje programa zahvaljujući niskoj ceni. Pouzdanost umrežavanja radnih stanica i PC računara u lokalne (*eng.* Local Area Networks - LAN) i globalne računarske mreže (*eng.* Wide Area networks - WAN) je postalo moguće tek sa razvojem operativnih sistema kao što UNIX i Windows NT. Performanse mreža radnih stanica zavise od tipa mreže koja se koristi. Tako lokalne mreže omogućuju brži protok informacija, ali smanjuju broj računara na kojima se aplikacija izvršava. S druge strane, Internet procesiranje dozvoljava pristup velikom broju računara koji mogu biti iskorišćeni, međutim međuprocorska komunikacija može biti spora.

### Superračunari i paralelni računari visokih performansi

Troškovi izrade superčunara su ograničavajući faktor, što ih čini praktično nedostupnima za širu klasu istraživača i programera. U slučaju paralelnih računara visokih performansi, cene su znatno manje mada im je pristup ograničen na uzak krug istraživača i programera na samo većim univerzitetima i istraživačkim centrima u Evropi, SAD i Japanu.

Zajedničke osobine superračunara i paralelnih računara su:

- Vrlo skupo računarsko vreme i visoki troškovi razvoja, testiranja i izvršavanja aplikacija;
- Izuzetno veliki napor u optimizaciji i efikasnom paralelnom izvršavanju, koji ulažu specijalizovani prevodioci datog programskog jezika ili sam programer. U suprotnom može doći do osetne degradacije performansi paralelnog računara;
- Tehnike optimizacije i poboljšanja performansi se obično mogu primeniti samo na tu klasu višeprocorskih računara.

## 1.5 Analiza Performansi

Prilikom projektovanja paralelnog programa korisno je utvrditi koje ćemo ubrzanje postići prilikom implementacije na paralelnom računaru. Vremenska analiza izvršenja paralelnog programa nam može dati informaciju o granici dostizanja visokih performansi programa, kao i o njegovom ponašanju prilikom povećanja broja procorskih čvorova na kojima se izvršava. Sledi da precizno predviđanje vremena izvršavanja

paralelnog programa koji razvijamo pomaže nam da utvrdimo da li ćemo postići ubrzanje ili usporenje.

### 1.5.1 Ubrzanje i Efikasnost

Svaki se paralelni algoritam delom sastoji od naredbi koje se izvršavaju serijski, zbog čega implementacije ovih paralelnih algoritama imaju tačku zasićenja posle koje se dalje ubrzanje ne može postići. **Ubrzanje (U)** predstavlja odnos vremena izvršavanja paralelnog programa (PP) na jednom i/ili više procesora, odnosno ako sa  $t_{S1}$  označimo vreme potrebno za serijsko izvršavanje (na jednom procesoru) i sa  $t_{P1}$  vreme paralelnog izvršavanja (na više procesora), ubrzanje koje se dobija izvršavanjem PP-a na paralelnim računarima je:

$$U = \frac{t_{S1}}{t_{P1}}. \quad (1.2)$$

Majkl Kuin je na osnovu svog radnog iskustva u oblasti paralelnog programiranja, izdvojio tri vrste operacija koje se javljaju u paralelnim programima ([28]):

- a) Operacije koje se moraju izvršavati sekvencijalno;
- b) Operacije koje se mogu izvršavati paralelno;
- c) *Parallel overhead* (višak operacija): Višak operacije koje procesor izvršava prilikom komunikacije sa drugim procesorom, i koje ne moraju imati veze sa radom paralelnog programa.

Na osnovu uvedenih kategorija ukupno vreme izvršavanja programa na paralelnom računaru sa  $n$  procesora se računa pomoću sledeće formule:

$$t_{\text{ukupno}} = t_a + \frac{t_b}{n} + t_c, \quad (1.3)$$

gde je  $t_a$  vreme izvršavanja operacija iz sekvencijalnog dela programa,  $t_b/n$  vreme izvršavanja operacija paralelnog dela programa na jednom od  $n$  procesora i  $t_c$  vreme komunikacije između svih  $n$  procesora koje obuhvata i vreme čekanja procesora da dobije ili pošalje podatke.

Sekvencijalni program (program koji može da se izvrši samo na jednom procesoru) može da izvrši samo jednu operaciju u jedinici vremena. Zbog toga je ukupno vreme potrebno za izvršavanje sekvencijalnog programa jednako  $t_a + t_b$ , jer se ne zahteva komunikacija sa nekim drugim procesorom.

Paralelni program (tj. sekvencijalni program koji smo pripremili za paralelno izvršavanje) u sebi sadrži sekvencijalni deo, odnosno onaj deo programa koji se ne može razdvojiti na nezavisne delove koji bi se mogli izvršavati konkurentno. Na osnovu formule 1.3 se primeti da vreme izvršavanja sekvencijalnog dela paralelnog programa ne zavisi od broja procesa koji učestvuju u računu. Ovo predstavlja i najbitnije

ograničenje prilikom ubrzavanja rada programa, jer vreme izvršenja paralelnog programa nikada ne može biti manje od vremena izvršenja njegovog sekvencijalnog dela. Formulu 1.2 zato možemo napisati u sledećem obliku:

$$U \leq \frac{t_a + t_b}{t_a + \frac{t_b}{n} + t_c}. \quad (1.4)$$

Iz formule 1.4 vidimo da se sa dodavanjem novih procesora smanjuje računsko vreme paralelnog programa, ali se sa ovim povećava vreme komunikacije, što može da dovede da u određenom trenutku ukupno vreme izvršavanja programa počne da raste.

**Efikasnost** ( $E$ ) paralelnog programa predstavlja meru iskorišćenosti procesora. Računa se kao odnos ubrzanja koje se postiže i broja procesora ( $n$ ) koji se koristi:

$$E = \frac{U}{n} \leq \frac{t_a + t_b}{nt_a + t_b + nt_c}. \quad (1.5)$$

U nastavku ćemo definisati Amdalov zakon (*eng.* Amdahl's law) koji omogućuje teorijsko određivanje maksimalnog ubrzanja našeg paralelnog programa ([28]).

### 1.5.2 Amdalov zakon

Amdalov zakon formalizuje izraz za ubrzanje prilikom izvršavanja programskih zadataka upotrebom većeg broja procesora, odnosno, tvrdi da je ubrzanje programa na višejezgarnim i višeprocorsnim sistemima ograničeno njegovim sekvencijalnim delom ([1]). Ovaj zakon se često koristi u praksi za teorijsko određivanje maksimalnog ubrzanja.

Da bi izveli Amdalov zakon iskoristićemo formulu za ubrzanje 1.4. Vreme komunikacije između procesorskih jedinica može biti i nula ( $t_c \geq 0$ ), te za ubrzanje dobijamo:

$$U \leq \frac{t_a + t_b}{t_a + \frac{t_b}{n} + t_c} \leq \frac{t_a + t_b}{t_a + \frac{t_b}{n}}. \quad (1.6)$$

Neka je  $S = \frac{t_a}{t_a + t_b}$  sekvencijalni deo izračunavanja (deo programskog zadatka koji nije moguće paralelizovati) i  $1-S$  deo programa koji se izvršava paralelno. Tada je maksimalno ubrzanje ( $U_{\max}$ ) koje se postiže na  $n$  procesora predstavljeno sledećom formulom:

$$U_{\max} \leq \frac{1}{S + \frac{1-S}{n}}$$

Ukoliko  $n$  teži beskonačnosti, maksimalno ubrzanje će iznositi  $1/S$ .

Primeru radi, ukoliko se samo 10% od programa ne može paralelizovati tada ubrzanje na višeprocorskim računarima, ne može biti veće od 10 puta, bez obzira na povećanje broja procesora.

Amdalov zakon pretpostavlja da je cilj paralelizacije minimizacija vremena izvršavanja, te iz tog razloga zanemaruje vreme komunikacije između procesa na osnovu čega se procenjuje vrednost ubrzanja. Zbog ovakvog nedostatka Amdalovog zakona Karp i Flat (Karp, Flatt) su predložili drugi način za dobijanje vrednosti ubrzanja paralelnih programa.

### 1.5.3 Karp-Flat metrika

Karp-Flat metrika se zasniva na eksperimentalno određenom serijskom delu programa  $e$  koji je jednak:

$$e = \frac{t_a + t_c}{t_a + t_b}. \quad (1.7)$$

Eksperimentalno određen serijski deo  $e$  je izuzetno korisna mera jer može da nam da informacije o višku operacija koje procesor preuzima prilikom izvršavanja zadataka, što se zanemaruje prilikom jednostavnije vremenske analize.

Neka je  $t_{\text{ukupno}}$  vreme definisano formulom 1.3. Uvrstićemo izraz za  $e$ , na osnovu čega dobijamo:

$$t_{\text{ukupno}} = (t_a + t_b)e + (t_a + t_b)\frac{1-e}{n}. \quad (1.8)$$

Zakon nam daje: Za ubrzanje  $u$  (1.4) koje se postiže na  $n$  procesora eksperimentalno određen serijski deo  $e$  je

$$e = \frac{\frac{1}{u} - \frac{1}{n}}{1 - \frac{1}{n}}. \quad (1.9)$$

Za program fiksne veličine efikasnost prilikom paralelnog računanja uglavnom opada sa rastom broja procesora. Korišćenjem Karp-Flat metrike moguće je utvrditi da li efikasnost opada zbog (1) ograničenosti programa za paralelizaciju, ili (2) rasta viška operacija, odnosno, onih operacija koje procesor izvršava nezavisno sa izvršavanjem paralelnog programa.

---

### Standardni interfejs za razmenu poruka, MPI

---

Zahvaljujući sve većoj upotrebi masivno paralelnih procesora<sup>1</sup> i klastera, standardni interfejsi za razmenu poruka kao što su PVM<sup>2</sup> i MPI<sup>3</sup> su postali popularni. Koriste se ne samo za pisanje konkurentnih programa, već i kao efikasan alat za paralelizaciju postojećih programa, a služe i kao biblioteka koja se koristi u drugim jezicima kao što je High Performance Fortran (HPF).

#### 2.1 Interfejs za prosleđivanje poruka - MPI

MPI je specifikacija standardne biblioteke funkcija za paralelni model prosleđivanja poruka. Na definisanju i razvoju MPI standarda je radio takozvani MPI Forum, grupa više od 68 istraživača iz raznih evropskih i američkih institucija ([33]). Proizvod dvogodišnjeg rada MPI Foruma je standard koji je danas podržan na velikom broju različitih paralelnih platformi, efikasan i sa programskim konstrukcijama relativno visokog nivoa. Prilikom formiranja, MPI standard je preuzeo konstrukcije raznih postojećih sistema (strana 18) koji su se dobro pokazali u praksi i ugradio ih u jedan pouzdan i efikasan sistem. Bez zahteva za kompatibilnošću sa starim verzijama omogućeno je projektovanje standarda koji će zadovoljiti i najzahtevnije prohteve. Početni dogovor oko MPI standarda je nastao aprila 1992. godine na radionici "Standardi za Prosleđivanje Poruka u Distribuiranim Sistemima" ([33]), da bi konačna verzija MPI

---

<sup>1</sup> *eng.* Massively Parallel Processors, MMP. Videti u dodatku A.6.1

<sup>2</sup> *eng.* PVM - Parallel Virtual Machine

<sup>3</sup> *eng.* MPI - Message Passing Interface

standarda 1.0 bila predstavljena na konferenciji "Supercomputing 1993" u Portlandu, USA.

MPI standard podržava sledeće elemente:

1. Pojedinačne međuprocesorske komunikacije (*eng.* point-to-point communications) raznih tipova. Uz veći broj sistemskih tipova, moguće je korišćenje i korisnički definisanih tipova.
2. Kolektivne operacije za međuprocesorsku komunikaciju nad sistemskim i korisnički definisanim tipovima.
3. Grupe procesa i manipulacija kontekstima, koji čuvaju zajedničke osobine grupa;
4. Korisnički definisane topologije koje mogu biti realne (odgovaraju fizičkim vezama) ili virtuelne.
5. Mogućnost svakog procesa da inicijalizuje, očitava ili menja komunikacioni kontekst svoje grupe procesa u toku izvršavanja.

Kako je MPI standard nezavisan od platforme i operativnog sistema na kojem se izvršava, svaka MPI implementacija mora dodatno da obezbedi ulazno/izlazne operacije i način izvršavanja aplikacija na datom konkretnom operativnom sistemu. Za prevođenje izvornog kôda i ispravljanje grešaka se koriste: standardni C, Fortran ili Java prevodioci i ostali alati posebno konfigurisani i podešeni za MPI aplikacije.

Osnovni model MPI-1 ne podržava koncept deljene memorije, dok je kod MPI-2 verzije modela ovaj koncept ograničen. I pored ovakvog nedostatka, MPI programi se redovno izvršavaju i na računarima sa deljenom memorijom.

### 2.1.1 MPI implementacije

Odmah nakon usvajanja MPI standarda pojavile su se i prve MPI implementacije. Razvoj velikog broja MPI implementacija za razne vrste paralelnih računara, od mreže radnih stanica do superračunara, dovelo je da MPI standard postane najprihvaćeniji standard za razvijanje paralelnih programa.

Neke od javno dostupnih implementacija su: MPICH, WMPI, HP MPI, MPI-FM (MPI on Fast Messages), LAM (Local Area Multicomputer), CHIMP/MPI, MPIAP, RACE-MPI. Postoji i veliki broj komercijalnih MPI implementacija kao što su proizvodi kompanija: MPI Software Tehnology, Alpha, HP, IBM, Hitachi, Silicon Graphics, Nec, Intel i drugi.

### 2.1.2 Projektovanje i implementacija MPI programa

MPI poseduje visok stepen fleksibilnosti, čemu ide u prilog činjenica da se MPI sistem može izvršavati na mreži heterogenih računara, odnosno skupu procesora ra-



zličitih arhitektura. Korisnik ne mora da vodi računa o tome da li se poruke razme-  
njuju između procesora istih ili različitih arhitektura, jer MPI automatski obavlja sve  
neophodne konverzije podataka i obezbeđuje odgovarajući protokol za komunikaciju.

MPI program može da se sastoji iz više instanci sekvencijalnog programa koji  
komuniciraju pozivima odgovarajućih funkcija MPI biblioteke, a ove funkcije mogu  
se svrstati u četiri grupe:

1. Funkcije koje vrše inicijalizaciju, upravljanje i okončavanje komunikacija;
2. Funkcije koje služe za komunikaciju između parova procesa;
3. Funkcije koje izvode operacije nad grupom procesa;
4. Funkcije za kreiranje proizvoljnih tipova podataka.

Pomoću navedenih MPI funkcija se na različitim tipovima arhitektura paralelnih  
računara uvek realizuju sledeći procesi:

- (i) Navođenjem direktiva operativni sistem raspoređuje kopije izvršnog pro-  
grama na svaki čvor klastera (procesore).
- (ii) Svaki procesor izvršava svoju kopiju programa.
- (iii) Različiti procesori paralelno (konkurentno) izvršavaju onaj deo programa u  
kome se koristi identifikacioni broj procesa.

U narednom delu navodimo primer programa pisan programskim jezikom FOR-  
TRAN 77 u kojem se pozivaju osnovne MPI funkcije.

### 2.1.3 Primer MPI programa

Ovde izlažemo kratak primer programa pisan u FORTRANU 77, koji poziva os-  
novne procedure MPI biblioteke, sa ciljem da ilustrujemo način "ugradnje" ovih MPI  
procedura u postojeće programe i izložimo njihove glavne uloge. Predstavljen je pro-  
gram 'mpitest':

```

1  program mpitest
2  include 'mpif.h'
3  integer ierr, rang, n      !n = ukupan broj procesa
4
5  call MPLINIT (ierr)
6  if (ierr.ne.MPLSUCCESS) then
7  write (*,*) 'An error occurred during MPI call. Abort.'
8  call MPLABORT (MPLCOMMWORLD, MPLERR_UNKNOWN, ierr)
9  endif
10 call MPLCOMMRANK (MPLCOMMWORLD, rang, ierr)
11 call MPLCOMM_SIZE (MPLCOMMWORLD, n, ierr)
12 write (*,*) 'I am ', rang, ' of ', n
13 call MPLFINALIZE(ierr)
14 end

```

Svaki MPI program počinje navođenjem predprocesorske direktive

```
include 'mpif.h'
```

gde fajl `mpif.h` sadrži definicije, makroe i prototipe funkcija potrebnih za prevođenje MPI programa.

Da bi se razlikovali od drugih standardnih naziva, sve funkcije i imena konstanti u MPI biblioteci sadrže prefiks `MPI_`.

### Funkcija `MPI_Init`

Funkcija koja se prva poziva iz MPI biblioteke je `MPI_Init`. Ona obezbeđuje sistemu da se izvrše sva podešavanja potrebna za poziv ostalih MPI funkcija. Poziv ove funkcije ne mora da bude prva naredba programa; čak nije potrebno pozvati je iz glavnog dela programa (`main` u C programima), ali je uvek bitno navesti je pre svih ostalih MPI funkcija.

`MPI_Init` se iz programa pisan FORTRAN-om poziva na sledeći način:

```
call MPIINIT ( ierr )
```

a iz C programa:

```
ierr = Mpi_Init();
```

Skrećemo pažnju da se inicijalizacija u programu može izvršiti samo jednom.

Svaka MPI funkcija poziva najmanje jedan argument, `ierr`, koji sadrži informacije o greškama. Kada je `ierr` jednaka predefinisanoj celobrojnoj konstanti `MPI_SUCCESS` znači da nije došlo do greške prilikom poziva MPI funkcije; u suprotnom se izvršenje programa prekida pozivom funkcije `MPI_Abort`.

### Funkcije `MPI_Comm_rank` i `MPI_Comm_size`

Posle inicijalizacije MPI funkcija svaki aktivan proces postaje član komunikatora koji definišemo pomoću `MPI_COMM_WORLD`. *Komunikator* je apstraktan objekat koji se brine o okruženju za razmenu poruka među procesima. `MPI_COMM_WORLD` je standardni komunikator MPI biblioteke. Ukoliko je potrebno korisnik može sam kreirati komunikatore deljenjem procesa na nezavisne komunikacijske grupe, pri čemu treba uzeti u obzir da procesi mogu komunicirati samo u slučaju kada dele isti komunikator.

Svaki proces jednog komunikatora dobija jedinstveni broj `id` (u primeru opisano parametrom `rang`). Kako su procesi jednog komunikatora jedinstveno određeni identifikacionim brojem, za  $p$  procesa svaki proces će imati jedinstveni identifikacion broj (`rang`) sa vrednošću od 0 do  $p-1$ . Na osnovu vrednosti `id`, procesori mogu da odrede koji deo programa je potrebno da izvršavaju.

Pozivom funkcije `MPI_Comm_rank` se svakom procesu dodeljuje jedinstveni broj, a funkcijom `MPI_Comm_size` se određuje ukupan broj procesa u komunikatoru (u primeru definisan parametrom `n`).

```
integer rank , ierr , n
call MPLCOMMRANK(MPLCOMMLWORLD, rang , ierr )
call MPLCOMM.SIZE(MPLCOMMLWORLD, n, ierr )
```

Ukoliko jedan proces pripada dvama različitim komunikatorima, njegov identifikacioni broj će za svaki komunikator biti drugačiji.

### Funckija `MPI_Finalize`

Posle poziva svih MPI funkcija, poziva se funkcija `MPI_Finalize` koja omogućuje sistemu da oslobodi svoje resurse (kao što je memorija) koji su bili rezervisani prilikom izvršavanja MPI programa.

### Tipovi podataka

Tipovi podataka, koji se prenose metodama interfejsa za slanje poruka (MPI), su uglavnom saglasni sa tipovima podataka u FORTRAN-u 77/90 za koje smo dali pregled u dodatku A.4.

### Interprocesorska komunikacija

Kod sistema sa distribuiranom memorijom, svaki procesor pristupa svojoj memoriji, odnosno među procesorima ne postoji deljena memorija. Jedini način da proces jednog računara pristupi memoriji drugog računara je eksplicitno slanje poruke. MPI obezbeđuje dve osnovne funkcije za razmenu podataka: `MPI_Send` i `MPI_Recv`.

Dajemo definiciju ovih funkcija u FORTRAN programskom jeziku:

```
MPLSEND (sendbuf , count , datatype , destinationRank , tag , comm , ierr )
MPLRECV (recvbuf , count , datatype , sourceRank , tag , comm , stat , ierr )
```

gde su:

- **sendbuf** - adresa prvog elementa koji se šalje
- **recvbuf** - adresa prvog elementa koji prima podatak
- **count** - broj elemenata koji se šalje (1 ukoliko se radi o primitivnom podatku ili  $n$  ukoliko se šalje niz podataka)
- **datatype** - MPI tip podataka koji se šalje (`MPLINT`, `MPLREAL`...)
- **destinationRank** - ID procesa kome se podatak šalje
- **sourceRank** - ID procesa od koga se podatak prima

- **tag** - identifikator koji se koristi da bi označio vrstu poruke, jer dva procesa mogu razmenjivati više različitih vrsta poruke; definisanje **tag**-a je ostavljeno programeru
- **comm** - komunikacioni domen; najčešće se koristi `MPI_COMM_WORLD` koji sadrži sve procese
- **stat** - sadrži informacije o primljenoj poruci, kao na primer sa kojeg čvora je podatak poslat, mada može da sadrži informacije i o tipu podataka koji se šalju

Pored navedenih procedura, MPI omogućuje korišćenje procedura koje pored slanja podataka istovremeno nad tim podacima vrše i osnovne aritmetičke operacije. Jednu takvu funkciju smo iskoristili u ovom radu, a njenu definiciju navodimo u zaključku.

Kako za paralelizaciju programa izračunavanja mape gravitacionih mikrosočiva nismo koristili funkcije `MPI_Send` i `MPI_Recv`, za dalje proučavanje čitaoce upućujemo na neke od knjiga i sajtova iz spiska referenci pobrojanih u ovom radu.

#### 3.1 Uvod

Gravitaciona sočiva su nebeski objekti čije gravitaciono polje savija zrake svetlosti koji prolaze pored njih, a koji potiču od nekog pozadinskog izvora. To znači da je neophodan uslov da neko telo bude gravitaciono sočivo da je njegova masa dovoljno velika da savije svetlost tokom svog prostiranja.

Prva pisana razmišljanja o uticaju mase na putanju svetlosti su se pojavila u poznatoj knjizi Isaka Njutna, "Optika" [15]. Sa ciljem da čitaocce usmeri na razmišljanje o ovom fenomenu Njutn se zapitao: "*Zar tela ne deluju na svetlost na nekom rastojanju i pomoću tog dejstva ne savijaju zrake, i zar nije to dejstvo najjače na najkraćem rastojanju?*". Ipak, bilo je potrebno dva veka da se efekat gravitacionih sočiva postavi na jače teorijske osnove što je postignuto primenom Opšte teorije relativnosti Alberta Ajnštajna. Od početka 20-og veka, a pogotovo u poslednjih 30 godina, zajedno sa tehnološkim razvojem astronomskih instrumenata gravitaciona sočiva su se razvila u jedan moćan alat za razna astrofizička i kosmološka istraživanja.

Savijanje svetlosti zavisi od mase sočiva, bez obzira da li je ono sastavljeno od tamne ili vidljive materije. Stoga je efekat gravitacionog sočiva (u nastavku GL, od engleskog *Gravitational lensing*) moćan alat za određivanje mase astronomskih tela sastavljenih od vidljive ili tamne materije. Zahvaljujući GL-u ostvarena su važna otkrića kao što je otkriće da se deo tamne materije u našoj galaksiji sastoji od objekata malih masa ([7],[8]); pronalazak jedne od najmanjih ekstrazolarnih planeta ([10]); otkriće velikih struktura u svemiru i pronalaženje direktnog empirijskog dokaza postojanja tamne materije ([11])([5]).

Interesantna osobina kod gravitacionih sočiva je da se ponašaju kao prirodni teleskopi, koji ponekada mogu biti moćniji od bilo kog teleskopa napravljenog ljudskom rukom. Zahvaljujući ovakvoj prirodi gravitacionih sočiva otkrivene su neke od najudaljenijih galaksija ([4],[5]).

## 3.2 Gravitaciono savijanje svetlosti

### 3.2.1 Njutnova i Ajnštajnova teorija

Pod pretpostavkom da se svetlost ponaša kao snop čestica <sup>1</sup> njeno savijanje može se odrediti pomoću Njutnove teorije gravitacije. Njutnova teorija tvrdi da je sila gravitacije  $\vec{F}$  koja deluje na česticu proporcionalna masi čestice ( $m$ ), te je gravitaciono ubrzanje

$$a = \frac{F}{m} = -G \frac{M}{r^2}, \quad (3.1)$$

gde je  $M$  masa centralnog objekta, a  $r$  rastojanje od njega, nezavisno od  $m$ . Posledica ovog zakona je da je trajektorija čestice u gravitacionom polju nekog tela nezavisna od njene mase i da za date početne vrednosti zavisi samo od brzine čestice.

Pre dvesta godina fizičari i astronomi su razmatrali uticaj gravitacionog polja na kretanje svetlosti. Tačnije, to su bili Džon Mičel (John Mitchell) 1784, i nešto kasnije Henri Kevendiš (Henry Cavendish), dok je 1804. godine Johan fon Soldner (Johann von Soldner) objavio rad u kome je izložio mogućnost da će se zrak svetlosti pod uticajem gravitacionog polja tela mase  $M$  (npr. zvezde) saviti za ugao

$$\hat{\alpha} = \frac{2GM}{c^2\xi}, \quad (3.2)$$

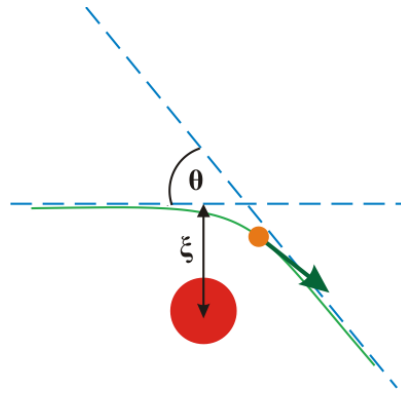
gde su  $G$  Njutnova konstanta gravitacije,  $c$  brzina svetlosti i  $\xi$  impakt parametar (videti sliku 3.1) nadolazećeg zraka svetlosti.

Godine 1915. godine je Ajnštajn za zrak svetlosti, koji prolazi pored tela mase  $M$  sa impakt parametrom  $\xi$ , iz jednačina Opšte teorije relativnosti (OTR), izračunao dvostruko veću vrednost za ugao savijanja ( $\hat{\alpha}$ ):

$$\hat{\alpha} = \frac{4GM}{c^2\xi} = 1''.75 \left( \frac{M}{M_\odot} \right) \left( \frac{\xi}{R_\odot} \right)^{-1}. \quad (3.3)$$

Iz poslednje formule se vidi da ugao savijanja zavisi od mase sočiva. Zbog toga masivni nebeski objekti, kao što su na primer galaksije, mogu saviti zrak svetlosti nekog pozadinskog izvora dovoljno jako da izazovu stvaranje njegovih višestrukih likova. Posmatrač će u tom slučaju registrovati zrake koji dolaze iz različitih pravaca, te će se izvor videti kao da postoji na više mesta.

<sup>1</sup> Detaljnije o modelu svetlosti u nastavku.



Slika 3.1: **Impakt parametar**  $\xi$ , tj. normalno rastojanje između pravca vektora brzine čestice i centra gravitacionog polja tela pored kojeg se čestica kreće.

Uticaj gravitacionih sočiva na prostiranje svetlosti dovodi do pojave sledećih interesantnih fenomena:

- (i) U slučaju sočiva velike mase, kao što su, npr. galaksije, dolazi do pojave višestrukih likova nekog pozadinskog izvora i takva sočiva se nazivaju **gravitaciona makrosočiva**.
- (ii) U slučaju manje masivnih sočiva, kao što su npr. zvezde, pojedinačni likovi pozadinskog izvora se ne mogu razdvojiti i tada dolazi samo do pojačanja intenziteta zračenja pozadinskog izvora. Takva sočiva se nazivaju **gravitaciona mikrosočiva**.
- (iii) Sočiva čija je masa nedovoljna da izazove bilo koji od prethodna dva efekta, mogu samo da dovedu do deformacija oblika pozadinskog izvora, nazivaju se **slaba gravitaciona sočiva**.
- (iv) Svetlosni zraci putuju različitim putanjama od izvora do posmatrača i zbog toga stižu do njega u različitim trenucima, što predstavlja tzv. efekat **kašnjenja svetlosti**.

Savijanje svetlosti možemo da zapazimo i u našoj relativnoj blizini. Na osnovu OTR-a svetlosni zrak koji prolazi blizu Sunčeve površine savije se za ugao od  $1.74''$  ([6]), što je i Artur Eddington (Arthur Eddington<sup>2</sup>) potvrdio u posmatranjima tokom totalnog pomračenja Sunca 1919. godine.

<sup>2</sup> Arthur Stanley Eddington, engleski fizičar i astronom

### 3.2.2 Otkrića

#### Prvo otkriće višestrukih likova (1979.)

Termin *sočivo* u kontekstu gravitacionog savijanja svetlosti je prvi put primenio Oliver Lodž<sup>3</sup> 1919-te, sa napomenom da se radi o sočivu bez žižne daljine. Godine 1924. Danijel Hvoljson<sup>4</sup> je razmotrio situaciju u kojoj se izvor nalazi tačno na liniji određenoj nekim masivnim objektom i posmatračem. Zaključio je da će posmatrač tada izvor registrovati kao prsten oko spomenutog masivnog objekta, a u situaciji u kojoj se posmatrač, sočivo i izvor ne nalaze tačno na istoj liniji, dolazi do formiranja dva lika pozadinskog izvora na suprotnim stranama tog masivnog objekta.

Zahvaljujući Hvoljsonu se postavilo pitanje da li se ovakvi fenomeni mogu posmatrati ukoliko savijanje svetlosti vrše zvezde iz naše galaksije. Iz ovog razloga je Albert Ajnštajn 1936. godine napisao rad u kome je zaključio da ukoliko zvezde imaju ulogu sočiva, da će ugaona razdvojenost između likova biti isuviše mala (reda mililučnih sekundi) da bi se detektovali. Međutim, neki su naučnici (među kojima je bio Fric Cviki (Fritz Zwicky)) krenuli u potragu za ovakvim pojavama. Cviki je 1937. godine objavio dva rada u kojima je razmatrao uticaj "vangalaktičkih maglina" (tj. galaksija) na veoma udaljene i sjajne objekte. U svojim radovima se bavio određivanjem mase maglina, a za jednu od procena mase je izračunao da će se formirani likovi nalaziti uglavnom na ugaonom rastojanju od 10", što je tada bilo moguće detektovati optičkim teleskopom (dobijeni rezultat je za red veličine veći od današnjih rezultata). Uvideo je da se ovim može dodatno testirati OTR i odrediti masa galaksija-sočiva ([39]). U drugom radu Cviki nastavlja istraživanje o uticaju GL-a, tražeći verovatnoću detektovanja udaljenog izvora koji se nalazi pod uticajem gravitacionog sočiva ([40]). Ovde je zaključio da bi se skoro svaki 400-ti udaljeni izvor trebalo da nađe pod uticajem gravitacionih sočiva, čime je otkrivanje vangalaktičkih sočiva učinio praktično izvesnim. Uprkos njegovim naporima, proučavanje gravitacionih sočiva je mirovalo sve do početka 60-tih godina 20-og veka kada je ovom oblašću počeo da se bavi Refsdal<sup>5</sup> u svom magistarskom radu iz 1962. godine za kojim su usledili pionirski radovi iz 1964-te. Iako je teorija smatrana neosporivom, ideja da postoje objekti koji su izuzetno sjajni i na velikim udaljenostima je smatrana nerealnom (što je već 1963. doživelo preokret otkrivanjem i identifikacijom kvazara 3C 273, a ubrzo su usledila otkrića i drugih kvazara). Ovo je iniciralo klasifikaciju nove populacije objekata koji su kompaktni, sjajni i na velikim udaljenostima. Petnaest godina posle prvog otkrića kvazara je proteklo pre nego što je detektovan uticaj gravitacionih sočiva na populaciju ovih objekata.

---

<sup>3</sup> *Sir Oliver Joseph Lodge*, engleski fizičar.

<sup>4</sup> *Daniil Abraamovič Hvoljson*, ruski naučnik jevrejskog porekla.

<sup>5</sup> *Sjur Refsdal*, norveški astronom.



Utica j gravitacionih sočiva na kvazare je prvi put registrovao Volš<sup>6</sup> 1979. godine. Volš je u okviru projekta identifikovanja radio-izvora pomoću optičkih teleskopa otkrio par kvazara 0957+561 A i B, na međusobnom ugaonom rastojanju od 6", istim crvenim pomacima ( $z = 1.41$ ) i sa identičnim spektrima ([34]). Iste godine nastaju i dva uređaja koji su imali značajnu ulogu u posmatranjima gravitacionih sočiva: CCD detektor i VLA (Very Large Array) radio-interferometar. Pomoću VLA je pokazano da se u radu Volša radilo o kompaktnim radio-izvorima sa sličnim radio-spektrima. Nedugo zatim je između ova dva lika kvazara detektovana galaksija na crvenom pomaku  $z = 0.36$  koja pripada malom jatu galaksija ([32], [37]). Danas znamo da na likove datog kvazara zapravo utiče to galaktičko jato.

Godinu dana posle Volšovog rada otkriveno je i gravitaciono sočivo PG 115+080, kod koga je na prvi pogled izgledalo da postoje tri lika, među kojima je jedan lik delovao sjajnije u odnosu na druga dva. Ubrzo je pokazano da se sjajniji lik u stvari sastoji od dva manja lika na ugaonom rastojanju od  $\sim 0''.5$ , što je bilo tada teško detektovati pomoću klasičnih (optičkih) teleskopa sa Zemlje ([30]).

Do danas je pronađeno oko stotinak sistema sa višestrukim likovima sa galaksijama u ulozi gravitacionog sočiva. Nekoliko GL-a je otkriveno sistematskim posmatranjima radio-izvora, dok je veći deo njih otkriven slučajno ([30]). Tokom 1990. godine realizovane su potrage za gravitacionim sočivima zahvaljujući svemirskom teleskopu Hubble ([22]), pregledom među 15 000 radio izvora (JVAS<sup>7</sup> i CLASS<sup>8</sup>) i pregledom među vrlo sjajnim kvazarima na velikim crvenim pomacima ([19], [9]).

### Džinovski sjajni lukovi

Godine 1986. dve grupe istraživača su nezavisno ukazale na postojanje izuzetno izduženih, zakrivljenih objekata između dva jata galaksija ([30]) (videti sliku 3.2).

Postavljene su razne hipoteze kojima je pokušano da se objasni postojanje lukova u jatu galaksija i sve su se pokazale netačnim osim jedne u sklopu koje je Paćinski iskoristio crvene pomake ([27]). Utvrđeno je da su crveni pomaci lukova daleko veći od crvenog pomaka jata, što je bio direktan dokaz da su lukovi u stvari pojačani<sup>9</sup> i izobličeni likovi neke normalne galaksije koja se nalazi iza galaktičkog jata. Najveći doprinos u proučavanju sjaja lukova i identifikaciji višestrukih likova je dao HST<sup>10</sup>.

---

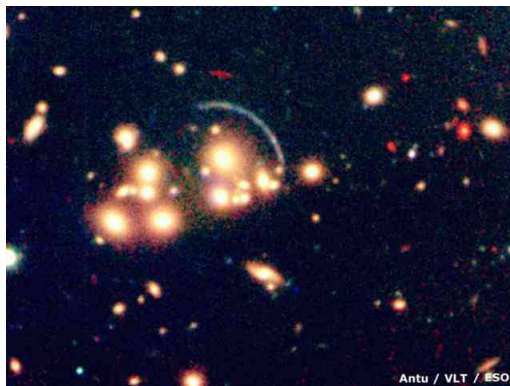
<sup>6</sup> *Dennis Walsh*, engleski astronom.

<sup>7</sup> *eng.* Jodrell Bank VLA Astrometric Survey.

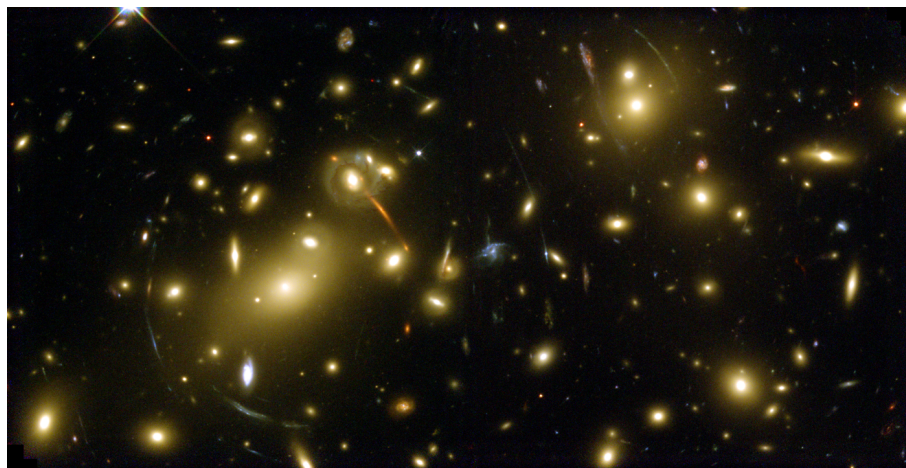
<sup>8</sup> *eng.* Cosmic Lens All Sky Survey.

<sup>9</sup> Pojačanje = *eng.* magnification.

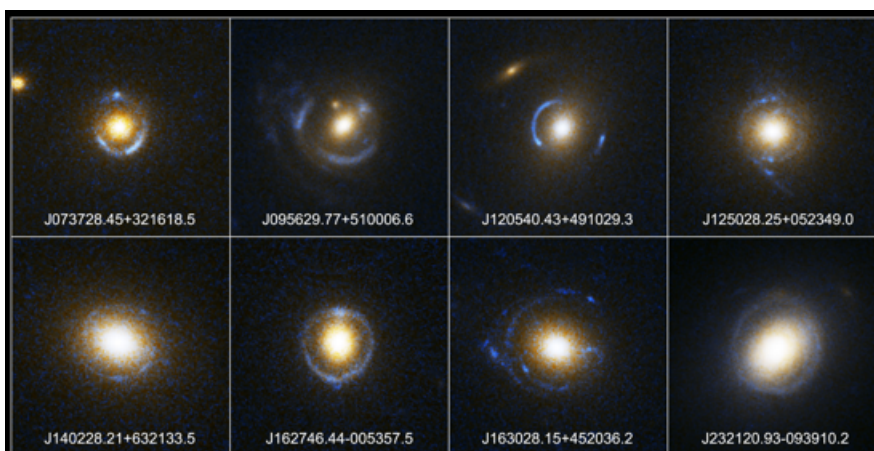
<sup>10</sup> HST (*eng.* Hubble Space Telescope).



*Slika 3.2:* Džinovski luk u jatu galaksija CI 2244-02. Luk se nalazi na crvenom pomaku  $z = 2.24$ . Veliko pojačanje koje prouzrokuje gravitaciono sočivo je kategorisalo ovu galaksiju kao najsjajniju galaksiju na crvenom pomaku većem od 2.



*Slika 3.3:* Na slici je prikazano jato galaksija Abel 2218 snimljeno pomoću WFPC2 kamere (eng. Wide-Field Planetary Camera 2) smeštene na Hابل teleskopu. Jato sadrži nekoliko višestrukih likova pozadinske galaksije. Zahvaljujući morfologiji lukova sa slike, omogućena je veoma detaljna konstrukcija modela za masu jata galaksija - Abel 2218.



Slika 3.4: Prikaz različitih Ajnštajnovih prstenova snimljenih pomoću Habl teleskopa (HST - Hubble Space Telescope).

## Prstenovi

Pronaći pozadinski izvor koji je savršeno poravnat sa nekim gravitacionim sočivom u odnosu na posmatrača na Zemlji je skoro nemoguć zadatak. U ovakvoj situaciji bi, prema teoriji gravitacionih sočiva, trebalo da vidimo tzv. *Ajnštajnov prsten*. Godine 1988. zahvaljujući optičkim posmatranjima detektovan je slab prstenasti objekat MG 1131+0456 ([13]). S obzirom na slab sjaj ovog objekta nije lako dokazati da se radi o efektu gravitacionog sočiva, mada su radio posmatranja godinu dana kasnije pomogla da se ova pojava tako okarakterise (Kochanek et al. 1989).

Nakon 1988. godine, usledila su razna druga otkrića Ajnštajnovih prstenova, a ovo ilustrujemo serijom fotografija na slici 3.4.

## Uticaj gravitacionih mikrosočiva na likove kvazara

Raspodela mase u galaksijama-sočivima nije ujednačena, s obzirom da jedan deo te mase čine zvezde. Ove zvezde utiču na (makro-)likove kvazara tako što stvaraju mikro-likove između kojih je tipična udaljenost nekoliko mikro-lučnih sekundi, što ih za današnja posmatranja čini nerazdvojivima. Na taj način dolazi do **efekta gravitacionog mikrosočiva**, odnosno pojačanja zračenja prilikom "prelaska" zvezde preko izvora. S obzirom da ni izvor ni sočivo, a pogotovo posmatrač, nisu stacionarni, pojačanje izvora se menja tokom vremena sa karakterističnom vremenskom skalom reda deset godina ili manje (u zavisnosti od oblasti spektra zračenja). Kao što su to predvideli Čeng i Refsdal ([2], [3]), Paćinski ([25]), Kajser ([18]) i Šnajder i Vajs ([29]) efekat mikrosočiva bi trebalo da utiče na promene fluksa likova između kojih ne postoji korelacija. Godine 1989. ovakav efekat mikrosočiva je detektovan kod četiri lika kvazara QSO 2237+0305 ([14]).

## 3.3 Teorija gravitacionih sočiva

### 3.3.1 Karakteristike svetlosti

Da bi opisali svetlost, na koju primenjujemo Njutnovu fiziku ili OTR, na raspolaganju nam stoje sledeća tri modela:

- Model zraka
- Talasni model
- Čestični model.

Čestični model svetlosti se koristili prilikom rešavanja ugla savijanja primenom Njutnovih zakona. Za razliku od njega, model svetlosti koji se najčešće koristi za opis uticaja gravitacionih sočiva na kretanje svetlosti (ne ulazeći u prirodu svetlosti) je *model zraka*, koji je postao uobičajen u većini radova o GL-u, te ga i mi koristimo u ovom radu.

### 3.3.2 Kosmološke osnove

U kosmologiji se naučnici bave izvorima svetlosti i gravitacionim sočivima koji se nalaze na umereno ili izuzetno velikim crvenim pomacima. Iz tog razloga se najmanja skala na kojoj se vrše posmatranja predstavlja u kiloparsecima (kpc), dok je veliki broj gravitacionih sočiva detektovan na udaljenostima od čak nekoliko gigaparseka (Gpc).

S obzirom da se radi o kosmološkim udaljenostima, potrebno je izabrati pogodnu metriku radi konstrukcije modela gravitacionih sočiva. Za opis geometrije Univerzuma koristi se tzv. *Fridman-Lemetr-Robertson-Voker*<sup>11</sup> (Friedmann-Lemaître-Robertson-Walker) metrika. Ona opisuje homogen<sup>12</sup> univerzum koji se izotropno širi. U slučaju gravitacionih sočiva, pogodnije je koristiti uprošćene modele prostora u kojima bi njihov uticaj na kretanje svetlosti predstavili preko potencijala gravitacionog polja sočiva ( $\Phi$ ).

Jedan od načina uvođenja potencijala u takav prostor jeste izvođenje Njutnovih zakona gravitacije iz jednačina polja Opšte teorije relativnosti, pri čemu moraju da važe sledeći uslovi:

- Gravitaciono polje, koje izaziva savijanje zraka svetlosti, je "slabo" (važi uslov  $|\Phi|/c^2 \ll 1$ ),
- Brzina čestica ( $v$ ) u tom polju je mala,
- Gravitaciono polje sočiva je statično (ne menja se tokom vremena).

<sup>11</sup> Skraćeno: Fridman-Voker metrika

<sup>12</sup> Za Univerzum kažemo da je *homogen* na velikim skalama (većim od 10Mpc) na osnovu ravnomernosti koja se javlja među tipovima galaksija, gustini galaksija u jatima, hemijskom sastavu i dr.

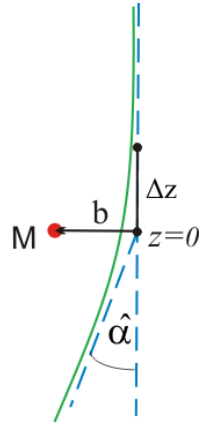
Na osnovu navedenih kriterijuma se iz jednačina OTR-a dobija graničan slučaj u kojem se nova metrika definiše pomoću Njutnovog potencijala  $\Phi$ , prostornih ( $d\vec{x}$ ) koordinata i vremenske koordinate ( $dt$ ). Ovakva metrika opisuje "skoro" ravan prostor, odnosno zakrivljen četvorodimenzioni prostor čija se (mala) razlika od ravnog prostora izražava preko Njutnovog gravitacionog potencijala  $\Phi$ . Za opis metričke se koristi interval prostor-vremena  $ds^2$  ([31])

$$ds^2 = -\left(1 + \frac{2\Phi}{c^2}\right)c^2 dt^2 + \left(1 - \frac{2\Phi}{c^2}\right)(d\vec{x})^2. \quad (3.4)$$

### 3.3.3 Ugao savijanja

#### Ugao savijanja i gravitacioni potencijal tačkastog sočiva

Na slici (3.5) je prikazan zrak svetlosti koji prolazi pored tela (tačkaste) mase  $M$ , a sa  $z_0 = 0$  tačka u kojoj se zrak najviše približava sočivu. Sa slike možemo da vidimo



Slika 3.5: Savijanje zraka pod uticajem tačkaste mase. Dolazeći zrak, sa impakt parametrom  $b$ , prolazi pored mase  $M$  i savija ga za ugao  $\hat{\alpha}$ . Nova putanja se aproksimira pomoću dve poluprave sa vrhom u tački  $z = 0$ .

prelazi onu putanju za koju je potrebno minimalno vreme (u slučaju GL-a odgovara tačka maksimuma).

Ukoliko u prostoru opisano metrikom (3.4) primenimo Fermaov princip i uslov  $ds = 0$  ([31]), za ugao savijanja zraka  $\hat{\alpha}$  ([24]) će važiti

$$\hat{\alpha} = \frac{2}{c^2} \int_{-\infty}^{+\infty} \nabla_{\perp} \Phi dz. \quad (3.5)$$

da se zrak najvećim delom savija unutar pojasa  $\Delta z \sim \pm b$  ( $\Delta z$  se meri u odnosu na  $z_0$ ). Veličina  $\Delta z$  je daleko manja od rastojanja sočiva, posmatrača i izvora, te većina sočiva ima osobinu *gravitaciono tankih sočiva* (detaljnije na strani 41).

Formula za ugao savijanja zraka svetlosti (3.3) se može izvesti iz jednačina polja OTR-a proučavanjem geodezijskih linija. Međutim, ovu formulu možemo izvesti i primenom *Fermaovog principa* iz geometrijske optike. Fermaov princip, ili princip ekstremalnog vremena, tvrdi da se svetlost prostire između one dve tačke svoje putanje na kojima je optička dužina puta (videti dodatak B.1) stacionarna, tj. minimum, maksimum ili prevojna tačka, odnosno da se prostire na taj način da prilikom prelamanja

U slučaju *tačkastog gravitacionog sočiva*<sup>13</sup> Njutnov potencijal gravitacionog sočiva ([24]) je definisan sledećim izrazom:

$$\Phi(b, z) = -\frac{GM}{(b^2 + z^2)^{1/2}}, \quad (3.6)$$

koji kada zamenimo u formulu (3.5), za ugao savijanja dobijamo

$$\hat{\alpha} = \frac{2}{c^2} \int \nabla_{\perp} \Phi dz = \frac{4GM}{c^2 b}. \quad (3.7)$$

Dobijena formula za ugao savijanja odgovara formuli izvedenoj iz Opšte teorije relativnosti (3.3).

### Ugao savijanja i masa proizvoljnog gravitacionog sočiva

U prethodnom delu smo formulisali vezu između ugla  $\hat{\alpha}$  i gravitacionog potencijala sočiva (3.5), odnosno tačkastog sočiva (3.7). U nastavku dajemo formule u kojima je data veza između ugla  $\hat{\alpha}$  i mase sočiva kada je masa proizvoljna.

Neka je sa  $m$  označena masa sočiva, odnosno suma masa pojedinačnih komponenti sočiva, a sa  $\rho(\vec{r})$  njena gustina. Za manje komponente sočiva ćemo uzeti da im je zapremina  $dV$  i masa  $dm = \rho(\vec{r})dV$ . Trajektoriju zraka svetlosti, koji prolazi pored ovakvog sočiva opisujemo vektorom  $(\xi_1(\lambda), \xi_2(\lambda), z(\lambda))$  u koordinatnom sistemu u kome se zrak iz izvora kreće duž ose  $z$  i gde je sa  $\lambda$  uveden afini parametar. U realnom slučaju zrak će se u blizini sočiva, pod uticajem gravitacionog polja, kretati po krivoj liniji (videti sliku 3.5). Ako je ugao savijanja mali onda trajektoriju zraka možemo da aproksimiramo pravom linijom, te takvo sočivo zovemo *gravitaciono tankim sočivom*.

Tada je impakt vektor zraka  $(\xi(\vec{\lambda}) \approx \vec{\xi})$  nezavisan od afinog parametra  $\lambda$ , te se može predstaviti sa  $\vec{\xi} = (\xi_1, \xi_2)$ . Impakt vektor je (relativno u odnosu na element mase  $dm$  opisan položajem  $\vec{r}' = (\xi'_1, \xi'_2, z')$ ) definisan vektorom  $\vec{\xi} - \vec{\xi}'$  i nezavisan je od  $z'$ , te za ukupan ugao savijanja ([30]) važi

$$\begin{aligned} \vec{\alpha}(\vec{\xi}) &= \frac{4G}{c^2} \sum dm(\xi'_1, \xi'_2, z') \frac{\vec{\xi} - \vec{\xi}'}{|\vec{\xi} - \vec{\xi}'|^2} \\ &= \frac{4G}{c^2} \int d^2 \xi' \int \rho(\xi'_1, \xi'_2, z') \frac{\vec{\xi} - \vec{\xi}'}{|\vec{\xi} - \vec{\xi}'|^2} dz' \end{aligned} \quad (3.8)$$

Kako pretposlednji faktor na desnoj strani jednačine 3.8 ne zavisi od parametra  $z'$ , to se integral po  $dz'$  može uprostiti uvođenjem *površinske gustine mase*  $\Sigma$

$$\Sigma(\vec{\xi}) = \int \rho(\xi_1, \xi_2, z) dz.$$

<sup>13</sup> *Tačkasto gravitaciono sočivo* predstavlja najjednostavniji model sočiva u kojem je masa koncentrisana u sferi, ravnomerno raspoređena oko centra te sfere.

Poslednji izraz predstavlja projekciju gustine mase na ravan normalnu na dolazeći zrak svetlosti. Odavde dobijamo da je ugao savijanja, kojeg prouzrokuje proizvoljna masa nekog gravitacionog sočiva, definisan kao dvodimenzioni vektor za kojeg važi

$$\vec{\alpha}(\vec{\xi}) = \frac{4G}{c^2} \int \Sigma(\vec{\xi}') \frac{\vec{\xi} - \vec{\xi}'}{|\vec{\xi} - \vec{\xi}'|} d^2\xi'. \quad (3.9)$$

### 3.3.4 Jednačina sočiva

Do sada smo predstavili zavisnost ugla savijanja od mase sočiva i od potencijala gravitacionog polja sočiva. Ove formule su osnovne za razumevanje prirode uticaja GL-a na kretanje zraka svetlosti. U nastavku definišemo parametre koji nam omogućuju da povežemo uticaj gravitacionih sočiva sa posmatračkim podacima.

Tipična situacija koja se izučava u oblasti gravitacionih sočiva je predstavljena slikom 3.6. Neka se gravitaciono sočivo **L** nalazi na crvenom pomaku  $z_l$ , odnosno, na rastojanju  $D_l$  od posmatrača **O**. Debelom (crvenom) linijom je označena putanja svetlosti od izvora **S** do posmatrača, pri čemu se izvor nalazi na crvenom pomaku  $z_s$ , odnosno, na udaljenosti  $D_s$  od posmatrača. Tačkastom linijom je aproksimirana putanja zraka svetlosti od izvora do posmatrača kada na taj zrak ne bi uticalo gravitaciono sočivo **L**.

U cilju opisivanja efekta gravitacionog sočiva uvodimo najpre optičku osu (pravu **p**) koja spaja položaj posmatrača **O** i centar mase gravitacionog sočiva **L**. Pomoću nje se dalje uvode dve pomoćne ravni:

- I) *Ravan izvora*, koja prolazi kroz izvor i normalna je na pravu **p**
- II) *Ravan sočiva*, koja prolazi kroz sočivo i normalna je na pravu **p**.

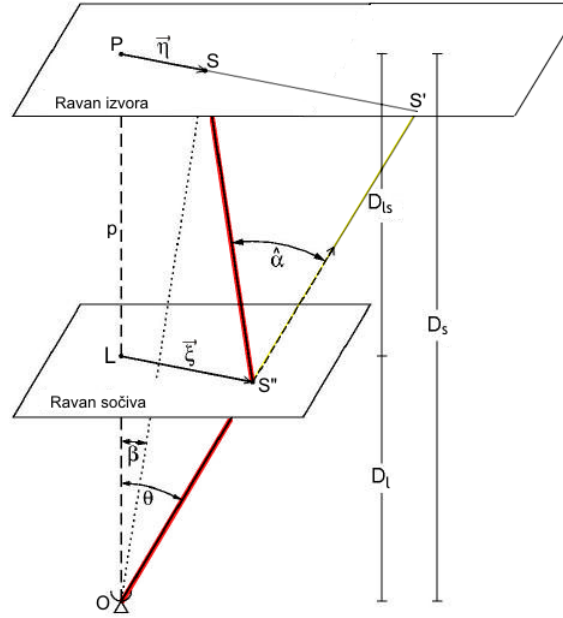
U pomoćnim ravnima izvora i sočiva se uvode merne referetne tačke tj. preseči ravnini sa optičkom osom. U ravni izvora je položaj izvora (**S**) definisan vektorom  $\vec{\eta}$ , koji se određuje u odnosu na presečnu tačku prave **p** sa ravni izvora, označenu sa **P**. Položaj lika izvora u ravni sočiva (**S''**) predstavili smo vektorom  $\vec{\xi}$ . Iz posmatračevog položaja **O** je sa  $\vec{\theta}$  označen ugao između optičke ose **p** i prave određene pravcem ka liku izvora (**OS''**), a sa  $\vec{\beta}$  ugao između optičke ose i pravca ka izvoru (**OS**).

Uzimajući u obzir da je ugao savijanja uglavnom mali, sa slike 3.6 izvodimo geometrijski uslov koji zovemo **jednačina gravitacionih sočiva**:

$$\vec{\eta} = \frac{D_s}{D_l} \vec{\xi} - D_{ls} \vec{\alpha}(\vec{\xi}) \quad (3.10)$$

pri čemu treba uzeti u obzir da je ova jednačina izvedena u tzv. aproksimaciji slabog gravitacionog polja. Rastojanja  $D_{l,s,ls}$  su kosmološka rastojanja za ugaone radijuse<sup>14</sup> u Fridmanovom kosmološkom modelu. U principu će važiti da je  $D_{ls} \neq D_s - D_l$ .

<sup>14</sup> eng. angular-diameter distance.



Slika 3.6: Savijanje zraka svetlosti koji se kreće od izvora  $S$  do posmatrača  $O$  pod uticajem gravitacionog sočiva  $L$

Za prelaz sa dekartovih na ugaone koordinate koristimo sledeće smene

$$\vec{\eta} = D_s \vec{\beta}, \quad \vec{\xi} = D_l \vec{\theta}, \quad (3.11)$$

te za jednačinu (3.10) će važiti

$$\vec{\beta} = \vec{\theta} - \frac{D_{ls}}{D_s} \vec{\alpha}(D_l \vec{\theta}) \equiv \vec{\theta} - \vec{\alpha}(\vec{\theta}), \quad (3.12)$$

gde smo sa  $\vec{\alpha}(\vec{\theta}) = \frac{D_{ls}}{D_s} \vec{\alpha}(D_l \vec{\theta})$  uveli *redukovan ugao savijanja*.

Jednačinu (3.12) tumačimo na sledeći način: izvor  $S$ , čiji je položaj opisan uglom  $\vec{\beta}$ , će na osnovu jednačine (3.12) imati lik  $S'$  koji će se videti pod uglom  $\vec{\theta}$ . Ukoliko za neko fiksirano  $\vec{\beta}$  jednačina (3.12) ima više rešenja, izvor će imati više likova, odnosno, gravitaciono sočivo će proizvesti višestruke likove.

U opštem slučaju, ugao savijanja je dvodimenzioni vektor (3.9). Razmotrimo zato specijalan slučaj sferno-simetričnog sočiva sa konstantnom površinskom gustinom  $\Sigma$  čija masa ima radijalnu raspodelu  $M(\xi)$ , za koju važi

$$M(\xi) = 2\pi \int_0^\xi \Sigma(\xi') \xi' d\xi' = \Sigma \pi \xi^2. \quad (3.13)$$

U ovom slučaju ugao savijanja svodimo na jednodimenzioni problem tako što koordinatni početak u kojem opisujemo jednačinu gravitacionih sočiva postavimo u



centar simetrije sočiva, zbog čega je

$$\hat{\alpha}(\xi) = \frac{4GM(\xi)}{c^2\xi} = \frac{4\pi G\Sigma}{c^2}\xi, \quad (3.14)$$

a za redukovani ugao važi

$$\alpha(\theta) = \frac{D_{ls}}{D_s}\hat{\alpha}(\xi) = \frac{4\pi G\Sigma}{c^2}\frac{D_l D_{ls}}{D_s}\theta, \quad (3.15)$$

gde je  $\xi = D_l\theta$ . Zamenom ovog izraza u jednačinu GL-a dobijamo linearnu zavisnost ugla  $\beta$  od ugla  $\theta$ .

Ovakav model gravitacionog sočiva nam omogućuje da uvedemo sledeću veličinu

$$\Sigma_{cr} = \frac{c^2}{4\pi G}\frac{D_s}{D_l D_{ls}} = \frac{0.35 \text{ g cm}^{-2}}{D/1\text{Gpc}} \quad (3.16)$$

koju nazivamo *kritična površinska gustina mase*, gde smo sa  $D$  označili *efektivno rastojanje*

$$D = \frac{D_l D_{ls}}{D_s}. \quad (3.17)$$

Veličina  $\Sigma_{cr}$  predstavlja karakterističnu vrednost površinske gustine sočiva koja razdvaja gravitaciona sočiva prema njihovom "slabom" ili "jakom" uticaju. Na primer, sočiva za koje je  $\Sigma > \Sigma_{cr}$  nazivamo *superkritičnim*, jer uglavnom proizvode višestruke likove (mada postoje i izuzeci).

## Ajnštajnov radijus

Bitna karakteristika gravitacionih sočiva je takozvani **Ajnštajnov radijus** (u ravni sočiva i u ravni izvora) kao i njemu odgovarajući **Ajnštajnov ugao**. Za izvođenje formule Ajnštajnovog radijusa koristimo sferno simetrični model sočiva sa proizvoljnom površinskom gustinom  $\Sigma$ . Jednačina gravitacionih sočiva je tada

$$\beta(\theta) = \theta - \frac{D_{ls}}{D_l D_s} \frac{4GM(\theta)}{c^2\theta}, \quad (3.18)$$

gde smo iskoristili drugi izraz formule ugla savijanja (3.15).

Gravitaciono sočivo će delovanjem na zrake svetlosti izvora koji se nalazi tačno na optičkoj osi ( $\beta = 0$ ), proizvesti lik u obliku prstena koji zaklapa ugao  $\theta_E$ . Ovaj ugao dobijamo kada uslov  $\beta = 0$  zamenimo u jednačini 3.18, odakle važi

$$\theta_E = \left[ \frac{D_{ls}}{D_l D_s} \frac{4GM(\theta_E)}{c^2} \right]^{1/2} \quad (3.19)$$

i njega nazivamo **Ajnštajnov ugao**. Iz formule 3.19 sledi da Ajnštajnov ugao zavisi ne samo od mase sočiva već i od rastojanja izvora od sočiva i posmatrača.

Na osnovu Ajnštajnovog ugla definišemo njemu odgovarajući **Ajnštajnov radijus**  $\xi_0$  u ravni sočiva ([17])

$$\xi_0 = D_l \theta_E = \left[ \frac{4GM(\theta_E) D_{ls} D_l}{c^2 D_s} \right]^{1/2}. \quad (3.20)$$

Često umesto Ajnštajnovog radijusa  $\xi_0$  u ravni sočiva koristimo njegovu projekciju na ravan izvora, koju ćemo označiti sa  $ERR$ <sup>15</sup> ([17]):

$$ERR = \frac{D_s}{D_l} \xi_0 = \left[ \frac{4GM(\theta_E) D_{ls} D_s}{c^2 D_l} \right]^{1/2}. \quad (3.21)$$

Ajnštajnovi radijusi nam omogućuju da uvedemo prirodnu skalu za ugao razdvajanja likova.

1. Kod detektovanih višestrukih likova tipični ugao razdvajanja likova iznosi  $2\theta_E$ .
2. Izvori na udaljenosti  $\leq \theta_E$  od optičke ose se nalaze pod uticajem jakih sočiva zbog čega će biti značajno pojačani, dok izvori koju su daleko van Ajnštajnovog radijusa će biti samo neznatno pojačani.
3. Srednja površinska gustina mase unutar Ajnštajnovog radijusa nekog sočiva odgovara njegovoj kritičnoj gustini  $\Sigma_{cr}$ .

## Potencijal gravitacionih sočiva

Jednačinom 3.12 je uvedeno preslikavanje  $\vec{\theta} \rightarrow \vec{\beta}$  ravni sočiva na ravan izvora, koje se lako određuje za proizvoljno  $\Sigma(\vec{\xi})$ . Kako je preslikavanje  $\vec{\theta} \rightarrow \vec{\beta}$  nelinearno, određivanje inverznog preslikavanja (tj. analitičkog izraza u kojem se ugao  $\theta$  predstavlja kao funkcija od ugla  $\beta$ ) je najčešće nemoguće. U tom slučaju se rešavanje inverzne jednačine gravitacionih sočiva svodi na primenu numeričkih metoda.

U cilju pronalazanja numeričkih rešenja u nastavku uvodimo potencijal gravitacionih sočiva koji je zbog svojih osobina pogodan za modeliranje GL-a, pogotovo kod određivanja pojačanja i deformacija likova nekog pozadinskog izvora.

Neka je sa  $\Psi(\vec{\theta})$  definisana projekcija Njutnovog potencijala  $\Phi$  na ravan sočiva na sledeći način

$$\Psi(\vec{\theta}) = \frac{D_{ls}}{D_l D_s} \frac{2}{c^2} \int \Phi(D_l \vec{\theta}, z) dz. \quad (3.22)$$

Potencijal  $\Psi$  nazivamo **potencijal gravitacionog sočiva**. Njegove glavne osobine su:

---

<sup>15</sup> eng. "Einstein Ring Radius"

- (I) Gradijent od  $\Psi(\vec{\theta})$  jednak je (*redukovanom*) uglu  $\vec{\alpha}$ , odnosno zahvaljujući formuli (3.5) važiće

$$\nabla_{\theta}\Psi = D_l\nabla_{\xi}\Psi = \frac{2}{c^2}\frac{D_{ls}}{D_s}\int\nabla_{\perp}\Phi(\vec{\xi},z)dz = \vec{\alpha}, \quad (3.23)$$

- (II) Laplasijan funkcije  $\Psi(\vec{\theta})$  je proporcionalan površinskoj gustini mase  $\Sigma$ :

$$\begin{aligned} \nabla_{\theta}^2\Psi &= \frac{2}{c^2}\frac{D_l D_{ls}}{D_s}\int\nabla_{\xi}^2\Phi dz = \frac{2}{c^2}\frac{D_l D_{ls}}{D_s}\cdot 4\pi G\Sigma \\ &= 2\frac{\Sigma(\vec{\theta})}{\Sigma_{cr}} \equiv 2\kappa(\vec{\theta}), \end{aligned} \quad (3.24)$$

Veličinu  $\kappa(\vec{\theta})$  zovemo **konvergens**<sup>16</sup> i ona predstavlja površinsku gustinu  $\Sigma$  izraženu u jedinicama kritične gustine  $\Sigma_{cr}$  ([23]). Jednačina  $\nabla_{\theta}^2\Psi = 2\kappa(\vec{\theta})$  predstavlja *dvodimenzionalnu Poasonovu jednačinu*.

Integracijom Poasonove jednačine (3.24) gravitacioni potencijal  $\Psi$  se izražava preko konvergensa ([30]), odakle sledi

$$\Psi(\vec{\theta}) = \frac{1}{\pi}\int_{\mathbb{R}^2}\kappa(\vec{\theta}')\ln|\vec{\theta}-\vec{\theta}'|d^2\theta'. \quad (3.25)$$

Na osnovu osobine (I) potencijala gravitacionog sočiva, redukovani ugao savijanja  $\alpha$  (gradijent funkcije  $\Psi$ ) se takođe izražava preko konvergensa  $\kappa(\vec{\theta})$ . Primenom karakteristika dvodimenzionog vektora  $\vec{\theta}$ <sup>17</sup>, sledi

$$\vec{\alpha}(\vec{\theta}) = \nabla\Psi = \frac{1}{\pi}\int_{\mathbb{R}^2}\kappa(\vec{\theta}')\frac{\vec{\theta}-\vec{\theta}'}{|\vec{\theta}-\vec{\theta}'|^2}d^2\theta', \quad (3.26)$$

što je ekvivalentno formuli (3.9) za ugao savijanja. Formula (3.26) nam daje direktnu vezu između ugla savijanja i mase sočiva koju određujemo iz posmatranja, te se i najčešće koristi u analitičkim i numeričkim modelima sočiva.

### Koeficijent pojačanja gravitacionog sočiva

Gravitaciona sočiva deluju na snopove zraka pozadinskog izvora tako što ih zakrivljuju pod različitim uglovima. Ovaj efekat najčešće (ako ne i uvek) dovodi do deformacija u obliku i veličini likova i najuočljiviji je kod izvora koji nemaju zane-marljivu prividnu veličinu.

<sup>16</sup> *eng.* convergence.

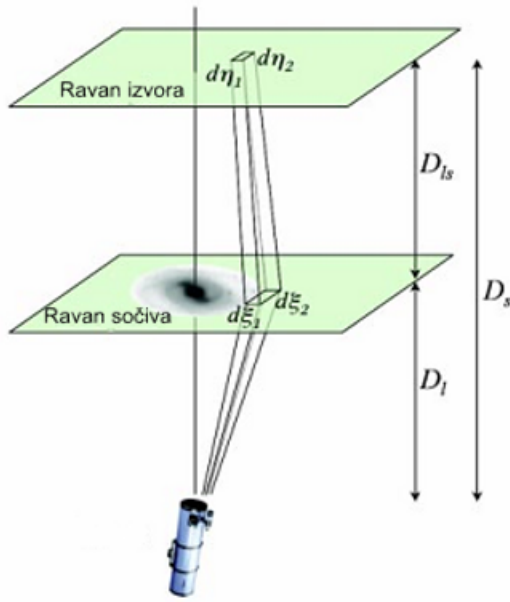
<sup>17</sup>  $\nabla\ln|\vec{\theta}| = \vec{\theta}/|\vec{\theta}|^2$  i  $\nabla^2\ln|\vec{\theta}| = 2\pi\delta_D(\vec{\theta})$ , gde je  $\delta_D$  Dirakova funkcija

Da bi opisali oblik likova uvodi se lokalno preslikavanje tačkaka ravni izvora na ravan sočiva, odnosno preslikavanje *malog izvora*<sup>18</sup> na *male likove* ([38]). Ovim se linearizuje primena jednačine gravitacionih sočiva.

Na osnovu Liuvilove (Liouville) teoreme<sup>19</sup> i činjenice da GL ne izaziva emisiju i apsorpciju fotona, sledi da se pod uticajem gravitacionih sočiva površinski sjaj (intenzitet) pozadinskog izvora ne menja. U tom slučaju, ako sa  $I^{(s)}(\beta)$  označimo dvodimenzionalnu raspodelu površinskog sjaja u ravni izvora, raspodela sjaja u ravni sočiva  $I(\vec{\theta})$  je određena sa ([30])

$$I(\vec{\theta}) = I^{(s)}[\vec{\beta}(\vec{\theta})]. \quad (3.27)$$

Određivanje vrednosti  $I(\vec{\theta})$  zovemo *mapiranje sočiva*, u kom slučaju se koristi navedeno lokalno preslikavanje, pod uslovom da je prividna veličina izvora manja od ugaone skale na kojoj se karakteristike sočiva menjaju ([30]).



Slika 3.7: Preslikavanje prostornih uglova

Na slici 3.7 je predstavljen uticaj gravitacionih sočiva na netačkasti izvor. Uglovi  $d\Omega_1^2$  i  $d\Omega_s^2$  predstavljaju prostorne uglove pod kojima se vide izvor i njegov lik, odnosno površi  $d\eta_1d\eta_2$  i  $d\xi_1d\xi_2$  u ravni sočiva i izvora. Ove površi su definisane tako da se, primenom jednačine GL-a, preslikavaju jedna u drugu, zbog čega prostorni uglovi  $d\Omega_1^2 = d\xi_1d\xi_2/D_l^2$  i  $d\Omega_s^2 = d\eta_1d\eta_2/D_s^2$  imaju različite vrednosti, te time i fluksevi zračenja izvora i likova.

Neka je sa  $F_0 = Id\Omega_s$  označen fluks izvora, a sa  $F = Id\Omega_1$  fluks lika. Promena fluksa zračenja usled uticaja gravitacionog sočiva zove se **pojačanje** ( $\mu$ ), i definisano je sa

<sup>18</sup> Izraz "mali izvor" se može odnositi na delove pozadinskog izvora.

<sup>19</sup> Liuvilova teorema tvrdi da duž snopa svetlosti važi  $n^2d\Omega^2dA^2 = \text{const}$ , gde je  $n$  indeks prelamanja,  $d\Omega$  prostorni ugao i  $dA$  površina poprečnog preseka snopa.

$$\begin{aligned} \mu &= \frac{F}{F_0} = \frac{d\Omega_l}{d\Omega_s} = \frac{d\xi_1}{d\eta_1} \frac{d\xi_2}{d\eta_2} \left(\frac{D_s}{D_l}\right)^2 = \frac{d\theta_1}{d\beta_1} \frac{d\theta_2}{d\beta_2} = \frac{1}{\det \frac{\partial \vec{\beta}(\vec{\theta})}{\partial \vec{\theta}} \frac{d\theta_1}{d\beta_1} \frac{d\theta_2}{d\beta_2}} \\ &= \left( \det \frac{\partial \vec{\beta}(\vec{\theta})}{\partial \vec{\theta}} \right)^{-1}. \end{aligned} \quad (3.28)$$

Poslednja jednakost je korektna ukoliko se preslikavanje  $\vec{\theta} \rightarrow \vec{\beta}$  može lokalno linearizovati. U tom slučaju, deformacije likova opisujemo pomoću Jakobijeve matrice

$$\begin{aligned} A(\vec{\theta}) &= \left[ \frac{\partial \vec{\beta}}{\partial \vec{\theta}} \right]_{2 \times 2} = \left[ \frac{\partial}{\partial \vec{\theta}} (\vec{\theta} - \vec{\alpha}(\vec{\theta})) \right]_{2 \times 2} = \left[ \frac{\partial}{\partial \vec{\theta}} (\vec{\theta} - \nabla \Psi) \right]_{2 \times 2} = \\ &= \left[ \delta_{ij} - \frac{\partial^2 \Psi(\vec{\theta})}{\partial \theta_i \partial \theta_j} \right]_{2 \times 2} = \left[ \delta_{ij} - \Psi_{ij}(\vec{\theta}) \right]_{2 \times 2}, \end{aligned} \quad (3.29)$$

gde  $\theta_i$  označava  $i$ -tu komponentu vektora  $\vec{\theta}$  u ravni sočiva. Formula 3.29 demonstrira da se elementi Jakobijeve matrice mogu predstaviti preko drugog izvoda funkcije potencijala  $\Psi$ .

Ako iz Jakobijeve matrice  $A$  isključimo njen izotropni deo  $\frac{1}{2}tr A(\vec{\theta})$

$$\begin{aligned} \left[ A(\vec{\theta}) - \frac{1}{2}tr A(\vec{\theta}) \cdot I \right]_{ij} &= \delta_{ij} - \Psi_{ij} - \frac{1}{2}(1 - \Psi_{11} + 1 - \Psi_{22})\delta_{ij} \\ &= -\Psi_{ij} + \frac{1}{2}(\Psi_{11} + \Psi_{22})\delta_{ij} \\ &\equiv \begin{bmatrix} -\frac{1}{2}(\Psi_{11} - \Psi_{22}) & -\Psi_{12} \\ -\Psi_{21} & \frac{1}{2}(\Psi_{11} - \Psi_{22}) \end{bmatrix}, \end{aligned} \quad (3.30)$$

dobija se simetrična matrica čiji je trag jednak nuli. Ovakvu matricu zovemo **šir-matrica** (eng. **shear**) i ona opisuje uticaj gravitacionog plimskog dejstva<sup>20</sup> koje izaziva deformaciju kod likova pozadinskog izvora ([23]).

Šir-matricom (3.30) je u ravni sočiva uvedena tenzorska veličina **šir**  $\vec{\gamma} = (\gamma_1, \gamma_2) \equiv \gamma_1 + i\gamma_2 = |\vec{\gamma}|e^{2i\varphi}$ <sup>21</sup> čije su komponente međusobno nezavisne i date sledećim formulama

$$\begin{aligned} \gamma_1(\vec{\theta}) &= \frac{1}{2}(\Psi_{11} - \Psi_{22}) \equiv \vec{\gamma} \cos[2\varphi(\vec{\theta})] \\ \gamma_2(\vec{\theta}) &= \Psi_{12} = \Psi_{21} \equiv \vec{\gamma} \sin[2\varphi(\vec{\theta})]. \end{aligned}$$

<sup>20</sup> Gradijent gravitacionog polja

<sup>21</sup> Između simetričnih,  $2 \times 2$  matrica (čiji je trag jednak nuli), i skupa kompleksnih brojeva postoji injektivno preslikavanje.

Sopstvene vrednosti šir-matrice (skraćeno šira) su

$$\pm \sqrt{\gamma_1^2 + \gamma_2^2} = \pm \gamma, \quad (3.31)$$

te će za rotacioni ugao  $\varphi$  važiti

$$\begin{bmatrix} \gamma_1 & \gamma_2 \\ \gamma_2 & -\gamma_1 \end{bmatrix} = \gamma \begin{bmatrix} \cos(2\varphi) & \sin(2\varphi) \\ \sin(2\varphi) & -\cos(2\varphi) \end{bmatrix}, \quad (3.32)$$

čime opisujemo orijentaciju šira kao i uticaj šira na zrake svetlosti<sup>22</sup> ([23]). Ovo je bitno za razmatranje uticaja gravitacionih mikrosočiva s obzirom da je koordinatni sistem, u kojem se opisuje jednačina mikrosočiva, orijentisan u saglasnosti sa orijentacijom šira.

Za izotropni deo Jakobijeve matrice važi izraz

$$\begin{aligned} \frac{1}{2} \text{tr} A(\vec{\theta}) &= \left[ 1 - \frac{1}{2} (\Psi_{11} + \Psi_{22}) \right] \delta_{ij} \\ &= \left( 1 - \frac{1}{2} \nabla^2 \Psi \right) \delta_{ij} = (1 - \kappa) \delta_{ij}. \end{aligned} \quad (3.33)$$

Zbir izotropnog (3.33) i neizotropnog (3.30) dela matrice  $A$  daje novi oblik Jakobijevoj matrici

$$\begin{aligned} A &= \begin{bmatrix} 1 - \kappa - \gamma_1 & -\gamma_2 \\ -\gamma_2 & 1 - \kappa + \gamma_1 \end{bmatrix} \\ &= (1 - \kappa) \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} - \gamma \begin{bmatrix} \cos(2\varphi) & \sin(2\varphi) \\ \sin(2\varphi) & \cos(2\varphi) \end{bmatrix}, \end{aligned} \quad (3.34)$$

koje nam dalje omogućuje da jednačinu sočiva definišemo za lokalno preslikavanje ([23]).

Neka je sa  $\vec{\theta}_0$  označen položaj proizvoljne tačke lika u ravni sočiva, koji odgovara tački izvora  $\vec{\beta}_0 = \vec{\beta}(\vec{\theta}_0)$ . Primenom lokalno linearizovane jednačine sočiva i jednačine 3.27, za raspodelu površinskog sjaja u ravni sočiva ([30]) će važiti

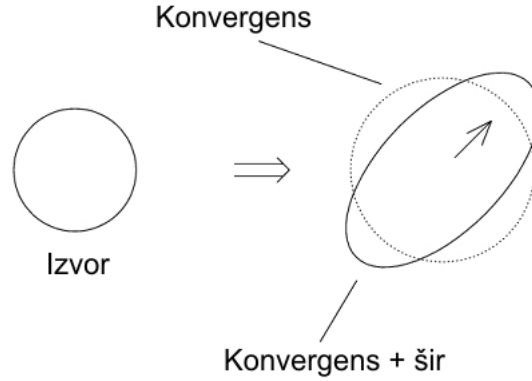
$$I(\vec{\theta}) = I^{(s)} [\vec{\beta}_0 + A(\vec{\theta}_0) \cdot (\vec{\theta} - \vec{\theta}_0)]^{[23]} \quad (3.35)$$

gde su sa  $\theta$  označene tačke u ravni sočiva koje se nalaze u neposrednoj okolini tačke  $\theta_0$ . Jednačina 3.35 ukazuje na to da će slike izvora sa kružnim izofotama<sup>24</sup> postati

<sup>22</sup> Ugao  $\varphi$  zove se *faza*, a faktor 2 je iskorišćen da bi podsetio da su komponente šira elementi  $2 \times 2$  tenzora, a ne vektora.

<sup>23</sup> Koristi se *Lagranževa teorema* srednje vrednosti za funkcije jedne promenjive.

<sup>24</sup> *Izofote* su linije koje spajaju tačke istog intenziteta.



Slika 3.8: Uticaj konvergenca i šira na sferno-simetričan izvor.

elipse, pod uslovom da su  $\kappa$  i  $\gamma$  ne-nula vektor i tenzor. Elipse su određene svojim poluosama

$$a = \frac{r}{1 - \kappa - \gamma}, \quad b = \frac{r}{1 - \kappa - \gamma},$$

gde je  $r$  poluprečnik izvora ([23]).

Bitna posledica uticaja gravitacionih sočiva je pojačanje koje opisujemo *tenzorom pojačanja* (matrica  $M$ ), za koji važi

$$M(\vec{\theta}) = A^{-1}, \quad (3.36)$$

Tenzor pojačanja  $M$  definiše lokalno preslikavanje tačaka ravni izvora na ravan sočiva. Kao što smo spomenuli, iz odnosa flukseva lika i izvora se određuje **koeficijent pojačanja**  $|\mu(\vec{\theta}_0)|$ , te znajući da se fluks ukupnog zračenja računa kao integral površinskog sjaja 3.35, za mali izvor će važiti

$$\mu = \det M = \frac{1}{\det A} = \frac{1}{(1 - \kappa)^2 - |\gamma|^2}. \quad (3.37)$$

Na osnovu formula (3.37) i (3.34) se može utvrditi do kojih sve deformacija dolazi u obliku i veličini likova. Tako u jednačini (3.34), prvi član  $(1 - \kappa)$  utiče samo na promenu u veličini lika, dok drugi član utiče na promenu oblika lika u odnosu na oblik izvora (videti sliku 3.8). Odavde se izvodi zaključak da deformaciju u obliku likova vrše plimska dejstva gravitacionog polja sočiva (opisana tenzorom  $\gamma$ ), a pojačanje (odnosno promenu u veličini) izaziva izotropno fokusiranje lokalnom gustinom materije opisano konvergencom  $\kappa$ , i anizotropno fokusiranje opisano širom  $\gamma$  odnosno *istezanje* u pravcu orijentacije šira. ([30], [23]).

Jakobijevom matricom  $A$  je, suštinski, opisana zavisnost oblika i veličine likova od oblika i veličine izvora. S obzirom da su gravitaciona sočiva uglavnom nepropusna

(i time osnovne informacije o izvoru nedostupne), formulu za pojačanje (3.37) ne možemo direktno primeniti. Iz ovog razloga se uvodi *relativna matrica pojačanja*

$$A_{ij} = A(\vec{\theta}_i)A^{-1}(\vec{\theta}_j), \quad (3.38)$$

koja opisuje linearno preslikavanje jednog lika u drugi ([30]).

Ukoliko je matrica  $A$  simetrična i ukoliko likovi imaju suprotne paritete <sup>25</sup> zbog čega je

$$\det A_{ij} < 0,$$

tada će likovi biti osno-simetrični. U opštem slučaju matrica  $A_{ij}$  nije simetrična ([30]).

### Kritične krive i kaustici

Iz Jakobijana preslikavanja  $\theta \rightarrow \beta$  se mogu izvesti pomoćne krive koje doprinose detaljnijem opisu uticaja gravitacionih sočiva analizom položaja likova u odnosu na ove krive. U ovom delu rada ćemo kratko opisati metod primene kritičnih krivâ i kaustikâ kod uticaja GL-a.

Zatvorenu, glatku krivu u ravni sočiva koju dobijamo iz jednakosti ([30])

$$\det A(\vec{\theta}) = 0 \quad (3.39)$$

nazivamo *kritična kriva*. Primenom jednačine gravitacionog sočiva ona se preslikava u krivu ravni izvora nazvanu *kaustik*. Termin "kaustik" je preuzet iz optike i predstavlja geometrijsko mesto centara krivine talasnih površi svih zraka iz datog snopa. Kako zraci iz snopa dodiruju kaustik, on čini obvojnici datog snopa. U specijalnom slučaju snopa sa zracima koji imaju sferne talasne površi, geometrijsko mesto centara krivine je jedna tačka - žiža. Postojanje kaustika je uvek prouzrokovano zakrivljenošću talasnog fronta, kao u slučaju fronta odbijenih ili prelomljenih talasa ([17]).

Izdvajamo sledeće osobine ovih krivih ([30]):

1. Koeficijent pojačanja  $\mu = \frac{1}{\det A}$  formalno divergira na kritičnim krivama. S obzirom da ovo nije fizički moguće ovakav uslov tretiramo u smislu pojačanja fluksa.
2. Broj likova nekog udaljenog izvora zavisi od njegovog relativnog položaja u odnosu na kaustik. Razmotrimo kao primer sočivo koje proizvodi ugao savijanja koje teži nuli za veliko  $|\vec{\theta}|$  i ograničimo gornju vrednost ugla savijanja (pri čemu ćemo za trenutak zanemariti tačkasta sočiva). Ukoliko je  $|\vec{\beta}|$  veliko izvor će imati samo jedan lik na položaju  $\vec{\theta} \approx \vec{\beta}$ , a može da ima višestruke likove za manje impakt parametre. Preslikavanje  $\vec{\theta} \rightarrow \vec{\beta}$  je

---

<sup>25</sup> Pojačanje definisano jednačinom 3.37 može imati oba znaka, što zovemo *paritet* (eng. *parity*) lika.



lokalno inverzno pod uslovom da je  $\det A \neq 0$ . Ovo znači da se sa promenom položaja izvora broj likova ne menja, sve dok izvor ne pređe preko kaustika. Kada se izvor nađe u blizini ili na samom kaustiku, u blizini odgovarajućih kritičnih krivih će se par likova ili formirati ili nestati, zavisno od smera kretanja izvora preko kaustika.

Strana kaustika na kojoj se broj likova **povećava za dva** se zove *unutrašnja strana kaustika*. Izvor, koji se sa unutrašnje strane približava kaustiku, će pored drugih likova dobiti par izrazito pojačanih likova. Kako koeficijent pojačanja menja znak u ovom slučaju, sjajan par likova će imati suprotne paritete.

Kaustici, za razliku od kritičnih krivih, ne moraju biti glatke krive. Dokaz za to je dat u Dodatku B.2.1.

Na osnovu osobina kritičnih kriva, ravan sočiva se može podeliti na dva regiona:

1.  $\mu > 0$ : region pozitivnog pariteta
2.  $\mu < 0$ : region negativnog pariteta,

dok odgovarajući kaustici dele ravan izvora prema broju likova: kada god izvor pređe preko kaustika, broj likova se promeni za  $\pm 2$  ([30]).

Uslov  $\det A = 0$  označava da će bar jedan od dva sopstvena vektora Jakobijeve matrice  $A$  biti nula-vektor. Zbog ovoga će likovi, koji se nalaze na kritičnim krivama, biti izduženi u pravcu odgovarajućeg sopstvenog vektora, te lik kružnog izvora ponekada može biti veoma izdužen.

### 3.4 Modeli sočiva

Jedan od glavnih ciljeva teorije gravitacionih sočiva jeste modeliranje njihovog uticaja, u čijim se okvirima razmatraju različite konfiguracije položaja i oblika kod izvora i sočiva. Veliku prednost imaju analitički modeli sočiva zbog svoje jednostavnosti i lakog izvođenja izraza za neke od karakteristika GL-a (ugao savijanja, gravitacioni potencijal sočiva, koeficijent pojačanja i promene oblika likova).

Prilikom definisanja kritične mase  $\Sigma_{cr}$  i Ajnštajnovog radijusa  $\theta_E$ , u ovom radu smo iskoristili jednostavan analitički model u kojem smo za masu sočiva usvojili osnosimetričnu raspodelu. Dobijeni parametri iz analitičkog modela koristimo za rešavanje jednačina gravitacionih sočiva za složenije konfiguracije sistema gravitacionih sočiva, kada masa celokupnog sistema više ne može imati simetričnu raspodelu.

Kompaktni objekti kao što su planete, zvezde, crne rupe, MACHO<sup>26</sup> i drugi manji objekti se aproksimiraju takozvanim *tačkastim sočivom*. U slučaju izduženih sočiva

---

<sup>26</sup> Objekti sačinjeni od tamne materije (*eng.* Massive compact halo objects).

kao najjednostavniji model se koristi osno-simetrični. Međutim, u većini situacija osno-simetrični model nije dovoljno realističan, zbog čega je adekvatniji tzv. *eliptični model* ([30]).

S obzirom na veliki broj raznolikih konfiguracija sočiva i izvora, u nastavku se ograničavamo samo one primere koji nam pomažu u modeliranju sistema gravitacionih mikrosočiva.

### 3.4.1 Analitički modeli sočiva

Kao što smo naveli, najjednostavniji modeli su oni u kojima se pretpostavlja da je raspodela masa sočiva sferno-simetrična, odnosno projekcija na ravan sočiva osno-simetrična. Ovaj uslov nam dozvoljava da rešavanje jednačine gravitacionih sočiva svedemo na jedan nepoznat parametar. Uvedenjem kriterijuma za ostale parametre sočiva, npr. masu sočiva (gustina i raspored masa), realizuju se različite vrste osno-simetričnih modela, od kojih navodimo

1. Tačkasti i
2. Model izotermalne sfere

U nastavku koristimo tačkasti model gravitacionih sočiva, s obzirom da je od najvećeg značaja za detektovanje efekata gravitacionih mikrosočiva, dok je model izotermalne sfere pogodan u slučaju određivanja raspodele (tamne) materije galaksija i jata galaksija.

### 3.4.2 Tačkasti model sočiva

Tačkasti model sočiva nam omogućuje da masu sočiva aproksimiramo tačkastom masom  $M$ , koncentrisanom u nekoj početnoj tački ravni sočiva. Tada će iz definicije (3.13) za površinsku gustinu u ravni sočiva važiti

$$\Sigma(\vec{\xi}) = M\delta_L(\vec{\xi}), \quad (3.40)$$

gde je sa  $\delta_L$  označena Dirakova funkcija. Zamenom  $\Sigma$  u formulu (3.14) za ugao savijanja sledi

$$\vec{\alpha}(\vec{\xi}) = \frac{4GM}{c^2} \frac{\vec{\xi}}{|\vec{\xi}|^2}, \quad (3.41)$$

na osnovu čega jednačina tačkastog gravitacionog sočiva ima sledeći oblik

$$\vec{\beta} = \vec{\theta} - \frac{D_{ls}}{D_l D_s} \frac{4GM}{c^2} \frac{\vec{\theta}}{|\vec{\theta}|^2} = \vec{\theta} - \theta_E^2 \frac{\vec{\theta}}{|\vec{\theta}|^2}. \quad (3.42)$$

U drugom koraku smo iskoristili definiciju (3.3) za Ajnštajnov ugao  $\theta_E$  koji zavisi od mase sočiva  $M$  i rastojanja posmatrača od izvora i sočiva. Ukoliko bez gubitka opštosti

izaberemo da se izvor (opisan uglom  $\vec{\beta} = (\beta_1, \beta_2)$ ) nalazi na pozitivnoj strani  $\beta_1$ -ose, tada će se i lik tog izvora (opisan uglom  $\vec{\theta} = (\theta_1, \theta_2)$ ) nalaziti na  $\theta_1$ -osi, na osnovu čega jednačina sočiva postaje jednodimenziona, a položaje opisujemo jednodimenzionim uglovima  $\theta$  i  $\beta$ . Ukoliko uglove predstavimo u jedinicama Ajnštajnovog ugla  $\theta_E$ , svodeći ih na parametre  $x$  i  $y$

$$x := \frac{\theta}{\theta_E}, \quad y := \frac{\beta}{\theta_E},$$

jednačina gravitacionog sočiva (3.42) se svodi na kvadratnu jednačinu

$$y = x - \frac{1}{x}, \quad (3.43)$$

koja ima dva rešenja

$$x_{1,2} = \frac{1}{2} \left( y \pm \sqrt{y^2 + 4} \right) \quad (3.44)$$

odnosno,

$$\theta_{1,2} = \frac{1}{2} \left( \beta \pm \sqrt{\beta^2 + 4\theta_E^2} \right). \quad (3.45)$$

Rešenja  $\theta_{1,2}$  predstavljaju ugaone položaje likova u ravni sočiva. Zahvaljujući simetričnosti sočiva, likovi će se u ravni sočiva nalaziti na suprotnim stranama od sočiva<sup>27</sup> na liniji koja spaja centar sočiva i izvora, (slika 3.9) i za rešenja će važiti da je  $\theta_1 > \theta_E$ , a  $-\theta_E < \theta_2 < 0$ .

Sa udaljavanjem izvora  $S$  od sočiva (tj. rastom ugla  $\beta$ ) lik, opisan na položaju  $\theta_1$ , će se približavati izvoru, dok će drugi konvergirati ka položaju sočiva. Ugaono rastojanje likova jednako je

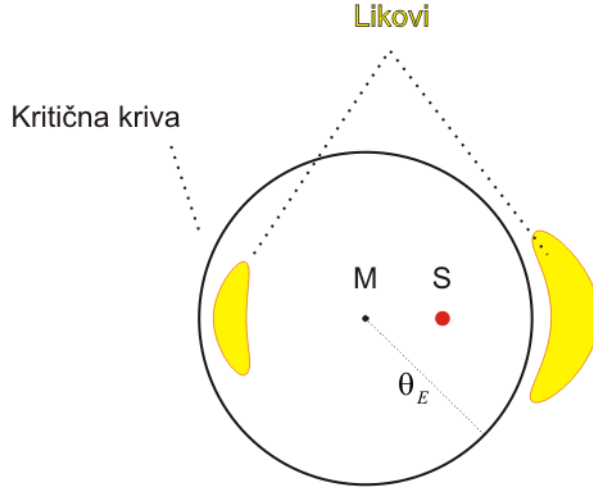
$$\Delta\theta = |\theta_1 - \theta_2| = 2\theta_E \sqrt{1 + \frac{\beta^2}{4\theta_E^2}} \geq 2\theta_E, \quad (3.46)$$

pri čemu je  $\Delta\theta$  u relevantnim situacijama samo neznatno veće od  $2\theta_E$ , jer će za  $\beta \geq \theta_E$  sjaj sekundarnog lika ( $\theta_2$ ) oslabiti ispod praga detekcije.

### Osno-simetrična sočiva

Za izvođenje koeficijenta pojačanja koje izaziva tačkasto sočivo, razmotrićemo model osno-simetričnih sočiva koje smo koristili prilikom definisanja Ajnštajnovog

<sup>27</sup>Ovaj podatak nam ukazuje na konfiguracije koje možemo da posmatramo u slučaju da je ugao razdvajanja dovoljno veliki.



*Slika 3.9:* Na slici je predstavljen efekat uticaja tačkastog gravitacionog sočiva ( $M \approx 10^{11} M_{\odot}$  na udaljenosti  $\sim 2.2 \cdot 10^9$  sv.g.) na izvor  $S$  kružnog oblika (veličine  $\sim 1''$ ). Generišu se dva lika sa različitih strana sočiva, pri čemu se lik (na istoj strani sočiva na kojoj je izvor) nalazi na većem rastojanju od sočiva nego drugi lik.

ugla. U opštem slučaju modela osno-simetričnih sočiva se masa sočiva može izraziti kao bezdimenziona veličina

$$m(\theta) = 2 \int_0^{\theta} d\theta' \theta' \kappa(\theta'), \quad (3.47)$$

dok smo sa  $\bar{\kappa}(\theta)$  uveli *srednju površinsku gustinu mase* koncentrisan unutar kruga poluprečnika  $\theta$  tako da važi

$$\bar{\kappa}(\theta) = \frac{m(\theta)}{\theta^2}. \quad (3.48)$$

Ukoliko iskoristimo poslednji izraz i definiciju ugla savijanja (3.14), za jednačinu osno-simetričnog gravitacionog sočiva će važiti

$$\vec{\beta} = [1 - \bar{\kappa}(|\vec{\theta}|)] \vec{\theta}. \quad (3.49)$$

Na osnovu definicije Jakobijeve matrice  $A$  (3.29) i diferenciranjem ugla  $\vec{\beta}$  po uglu  $\vec{\theta}$  sledi

$$A = (1 - \bar{\kappa})\mathcal{I} - \frac{\bar{\kappa}'}{|\vec{\theta}|} \begin{bmatrix} \theta_1^2 & \theta_1\theta_2 \\ \theta_1\theta_2 & \theta_2^2 \end{bmatrix}, \quad (3.50)$$

gde smo sa  $\mathcal{I}$  označili jediničnu dvodimenzionu matricu i  $\bar{\kappa}' \equiv d\bar{\kappa}/d\theta = 2[\kappa(\theta) - \bar{\kappa}(\theta)]/\theta$ . Poredeći sa konačnim izrazom Jakobijeve matrice (3.34), za šir osno-simetričnih sočiva će važiti

$$\gamma(\vec{\theta}) = [\kappa(\theta) - \bar{\kappa}(\theta)] e^{2i\varphi}, \quad (3.51)$$

te na osnovu (3.50) sledi

$$\det A = (1 - \kappa)^2 - |\gamma|^2 = (1 - \kappa)^2 - (\bar{\kappa} - \kappa)^2 = (1 - \bar{\kappa})(1 + \bar{\kappa} - 2\kappa). \quad (3.52)$$

### Koeficijent pojačanja tačkastog sočiva

Za dobijanje koeficijenta pojačanja tačkastog sočiva se koriste izvedene formule osno-simetričnih sočiva, tako što na osnovu (3.47) i (3.48) masa je definisana sa:  $m(\theta) = \theta_E^2$ , te za površinsku gustinu sada važi

$$\bar{\kappa}(\theta) = \left( \frac{\theta_E}{\theta} \right)^2 = \frac{1}{x^2}. \quad (3.53)$$

Na osnovu definicije (3.37) koeficijent pojačanja kojeg proizvodi tačkasto sočivo  $\mu$  je definisan kao

$$\mu = \frac{1}{\det A} = \frac{1}{(1 - \bar{\kappa}^2)} = \left( 1 - \frac{1}{x^4} \right)^{-1}. \quad (3.54)$$

Njega ćemo sada predstaviti kao funkciju od rastojanja  $y$  izraženog u jedinicama Ajnštajnovog radijusa, zamenom rešenja za  $x_{1,2}$  u poslednju jednačinu, čime dobijamo pojačanja za oba lika ( $\mu_1$  i  $\mu_2$ ):

$$\mu_{1,2} = \pm \frac{1}{4} \left[ \frac{y}{\sqrt{y^2 + 4}} + \frac{\sqrt{y^2 + 4}}{y} \pm 2 \right] = \pm \frac{1}{4} \left[ \frac{2y^2 + 4}{y\sqrt{y^2 + 4}} \pm 2 \right]. \quad (3.55)$$

Zbir pojačanja likova daje ukupno (apsolutno) pojačanje

$$\mu_p = |\mu_1| + |\mu_2| = \frac{2 + y^2}{y\sqrt{y^2 + 4}} = \frac{2\theta_E^2 + \beta^2}{\beta\sqrt{\beta^2 + 4\theta_E^2}}, \quad (3.56)$$

koje je jednako jedinici u slučaju gravitacionih mikrosočiva ([30]).

### Realan model GL-a. Kvadrupolna sočiva

Do sada smo se bavili jednostavnim modelima gravitacionih sočiva gde smo sočiva smatrali izolovanim, a za masu usvojili osno-simetričnu raspodelu. U realnom slučaju se za opisivanje mase sočiva koriste komplikovaniji modeli, kada je teško ili nemoguće naći analitički izraz. Rešenje nalazimo u numeričkom modeliranju, gde je opet potrebno usvojiti određene aproksimacije, kao na primer da je površinska gustina  $\kappa$  konstantna (npr. model mikrosočiva). Ovu aproksimaciju koristimo kod jednostavnijih klasa modela sočiva pomoću kojih se, pored usvojene osno-simetrične raspodele mase, opisuju spoljašnje perturbacije koje izazivaju plimska dejstva gravitacije objekata suviše malih masa da bi imali ulogu glavnih sočiva.

Bez obzira što masu kompaktnih objekata (zvezde ili galaksije) opisujemo osno-simetričnom raspodelom, rezultujući (ukupan) gravitacioni potencijal se ne povinuje

ovakvoj simetriji jer u opštem slučaju sočiva nisu izolovana (simetrija je poremećena uticajem tamne materije jata galaksija kojem galaksija-sočivo pripada ili joj je blizu) ([30]). Razmotrimo zato galaksiju koja pripada nekom jatu galaksija. Gravitaciono polje jata galaksije unosi dodatne perturbacije prilikom savijanja zraka koji prolazi pored galaksije-sočiva, zbog čega se njen uticaj mora proširiti uticajem potencijala celog jata. Analognu situaciju nalazimo kod uticaja zvezda udaljenih galaksija, gde simetriju tačkastog sočiva (zvezde) narušava plimsko dejstvo gravitacionog polja galaksije kojoj ta zvezda pripada (galaksije-domaćina). Zbog ovoga se ugao savijanja zraka svetlosti računa kao posledica uticaja celog sistema kojem razmatrano sočivo pripada.

Da bi se približili realnom modelu gravitacionih sočiva razmotrićemo model u kojem je raspodela materije glavnog sočiva (galaksije) osno-simetrična, u koji uključujemo i uticaj gravitacionog polja sistema-domaćina (jata galaksija). Ako sa  $p$  indeksiramo veličine vezane za perturbacije koje unosi ceo sistem, tada za jednačinu gravitacionog sočiva, koja opisuje ukupan uticaj glavnog sočiva (galaksije) i ostatka sistema (jata) ([30]), dobijamo

$$\vec{\beta} = [1 - \bar{\kappa}(|\vec{\theta}|)]\vec{\theta} - \begin{bmatrix} \kappa_p + \gamma_p & 0 \\ 0 & \kappa_p - \gamma_p \end{bmatrix} \vec{\theta} \quad (3.57)$$

Ukoliko uporedimo ovu jednačinu sa jednačinama GL-a za osno-simetrična sočiva, primetićemo da sada imamo drugi član koji se naziva *kvadrupolni član* jednačine. Usled spoljašnjih perturbacija, u (3.57) figurišu srednja površinska gustina materije jata  $\kappa_p$  i spoljašnji šir  $\gamma_p$ . Zato ovaj model sočiva nazivamo **kvadrupolna sočiva** ([17]).

Istu jednačinu koristimo ukoliko je glavno sočivo zvezda, a galaksija kojoj pripada unosi spoljašnje perturbacije. Ovakav slučaj koristimo kod modeliranja uticaja gravitacionih mikrosočiva koje ćemo sa većom pažnjom obraditi u nastavku.

---

### Gravitaciona mikrosočiva

---

Od prvog otkrića gravitacionih sočiva se uvidelo da pojedinačne zvezde galaksije (koja se ponaša kao makrosočivo) mogu značajno uticati na svojstva stvorenih likova ([34]). Da bismo opisali takav događaj moramo poznavati osobine uticaja jakih sočiva (prvenstveno, tačkastih sočiva) i perturbacije koje unose ostali objekti u njihovoj blizini. Gravitaciona mikrosočiva uglavnom predstavljaju sisteme malih sočiva koji su toliko blizu da se prilikom određivanja njihovog uticaja na kretanje zraka svetlosti sumiraju efekti koje svako mikrosočivo proizvodi. Iz tih razloga smo u prethodnim delovima rada dali jednu opštu sliku o sočivima, sa glavnim osvrtom na analitički model tačkastog sočiva sa osno-simetričnom raspodelom mase, u cilju isticanja onih fenomena pomoću kojih olakšavamo modeliranje ponašanja mikrosočiva.

Izveli smo formule za ugaonu razdvojenost likova predstavljajući ih kao funkcije od mase gravitacionog sočiva. Primenom ovih formula možemo reći da kada sočivo čini objekat velike mase (kao što su galaksije ili jato galaksija), njen uticaj na pozadinski izvor će prouzrokovati pojavu višestrukih likova. U slučaju gravitacionog sočiva manje mase (na primer zvezde), višestruki likovi se neće uočiti jer je ugao razdvajanja likova mali. Ipak, ono što posmatrač može da opazi je promena intenziteta sjaja. Takvo gravitaciono sočivo, s obzirom da ima sopstveno kretanje, izaziva promene u krivoj sjaja pozadinskog izvora i naziva se *gravitaciono mikrosočivo* ([17]). Gruba procena za objekte koji se ponašaju kao mikrosočiva je da imaju masu  $10^{-6} \leq m/M_{\odot} \leq 10^6$  ([35]). Pozadinski izvor je uglavnom kvazar, ali u principu efekat mikrosočiva se primenjuje i na druge udaljene objekte, kao na primer na udaljene supernove ili gama eksplozije.

Mikrosočiva mogu biti obične zvezde, braon patuljci, planete, crne rupe, molekularni oblaci ili neki drugi kompaktni objekti<sup>1</sup>. Najčešće se proučavaju mikrosočiva onih galaksija koja se ponašaju kao glavno (makro) sočivo. Međutim, mikrosočiva takođe mogu da se nalaze u jatu galaksija ili čak da se zamisle kao "slobodna" i da ispunjavaju međugalaktički prostor ([35]).

Razmatranjem prostorne raspodele kompaktnih objekata zvezdanih masa, lako se identifikuju dva astrofizička režima pod kojima se realizuju efekti gravitacionih mikrosočiva ([36]):

- [a] Kompaktni objekti Mlečnog puta ili u njegovom halou (ili bilo gde u Lokalnoj Grupi) se ponašaju kao sočiva koja utiču na zvezde u centru Galaksije (*eng.* Galactic Bulge), Malom i Velikom Magelanovom oblaku (LMC/SMC) ili Andromedinoj galaksiji (M31). Ovakvu situaciju možemo da nazovemo *uticaj zvezdanih mikrosočiva*. Krive sjaja koje nastaju pod uticajem ovakvih mikrosočiva su uglavnom jednostavne, jer se zvezde koje su nam bliže ponašaju prema pozadinskim zvezdama kao sočiva. Oblik krivih sjaja su jednoparameterska familija koja zavisi od impakt parametra izraženog u jedinicama Ajštajnovog radijusa ([26]).
- [b] Kompaktni objekti u udaljenim galaksijama (ili njihov halo) se ponašaju kao sočiva na još udaljenije kvazare. Ovakva situacija se uglavnom zove *uticaj mikrosočiva na kvazare* (*eng.* quasar microlensing). Krive sjaja koje se dobijaju pod uticajem mikrosočiva na kvazare su komplikovanije od onih koje izazivaju zvezdana mikrosočiva, jer se uticaj sočiva ne razmatra individualno. S obzirom na veliku gustinu objekata u galaksiji-sočivu, efekat savijanja koherentnog zraka svetlosti će proizvesti veliki broj različitih krivih sjaja ([25]).

## Detektovanje uticaja gravitacionih mikrosočiva

Ono što nam omogućuje detektovanje uticaja gravitacionih mikrosočiva na pozadinske izvore je relativno kretanje izvora, sočiva i posmatrača jednog u odnosu na druga dva. Zbog ovakvog ponašanja se konfiguracije mikrolikova menjaju tokom vremena, što se dešava i sa ukupnim (apsolutnim) pojačanjem. Zbog ovoga je efekat mikrosočiva dinamički fenomen.

---

<sup>1</sup> Objekat je kompaktan pod "uslovom" da je prividna veličina tog objekta manja ili jednaka od Ajštajnovog radijusa odgovarajućeg sočiva



Uticaje gravitacionih mikrosočiva možemo registrovati korišćenjem više tehnika pomoću kojih merimo promene različitih parametara ([36]):

- Fotometrijska tehnika: zasniva se na merenju promena prividne veličine,
- Spektroskopska tehnika: meri se promena oblika širokih emisionih linija<sup>2</sup> između različitih likova jednog kvazara
- Astrometrijska tehnika: zasniva se na merenju promene položaja kvazara.

Promena sjaja kvazara (ili nekog udaljenog izvora) se može objasniti na dva načina: ili je promena fizička karakteristika kvazara, ili posledica uticaja mikrosočiva. Međutim, u slučajevima kada raspoložemo sa više makro-likova moguće je izdvojiti uticaj gravitacionih mikrosočiva na promenu sjaja upotrebom *vremenskog zakašnjenja* do kojeg dolazi usled različitih dužina puta koje zraci svetlosti prelaze krećući se od likova do posmatrača (o vremenskom zakašnjenju detaljnije u dodatku B.2.3).

## 4.1 Model gravitacionih mikrosočiva

U modelu gravitacionih mikrosočiva znamo da značajan deo mase galaksija nije homogeno raspoređen, već se koncentriše u zvezdama. Efekat makrosočiva dovodi do formiranja dva ili više likova pozadinskog izvora za razdvojnim uglom od 1'' u slučaju kvazara. Usled "zrnastog" sastava makrosočiva svaki makro-lik će se sastojati od više mikro-likova sa razdvojnim uglom reda 1 mikro lučna sekunda. Zapravo, broj mikro-likova će približno biti jednak broju zvezda makrosočiva ([25]). Trenutnim posmatračkim tehnikama nije moguće razlučiti makro-likove na mikro, ali zahvaljujući relativnom kretanju između izvora, sočiva i posmatrača konfiguracije mikro-likova se menjaju, a sa time i apsolutno pojačanje koje se može izmeriti.

Za opis realističnih modela sistema mikrosočiva je potrebno uvesti veliki broj parametara. Jedan od razloga je struktura sistema mikrosočiva, odnosno vrste materije koje čine sistem mikrosočiva ([25]). Zato se, recimo, uvode različite definicije površinske gustine mase u zavisnosti od vrste materija (za međuzvezdani gas i neutrina ili "tamne" čestice i zvezde), minimalne i maksimalne mase zvezda itd.

---

<sup>2</sup> *eng.* shape of a broad emission line

### 4.1.1 Numerički metod

Spomenuli smo da se analitički izrazi za parametre gravitacionih sočiva definišu za specijalne slučajeve. Zbog toga se u modelima sočiva najčešće primenjuju numeričke metode rešavanja jednačine gravitacionih sočiva. Konkretnije, razlozi primene numeričkih metoda možemo razvrstati prema sledećim kriterijumima:

1. Nelinearnost jednačine gravitacionog sočiva, to jest, činjenica da su položaji lika i izvora povezani nelinearnom funkcijom;
2. Kada za analitički model gravitacionih sočiva tražimo detaljnije simulacije;
3. U slučaju kada radimo sa velikim brojem podataka.

Numeričke metode se najčešće koriste prilikom određivanja raspodele mase gravitacionih sočiva, mada su razvijene i tehnike pomoću kojih možemo da odredimo oblik likova slabih pozadinskih galaksija i signal gravitacionog šira.

Prilikom određivanja ukupnog uticaja sistema mikrosočiva na udaljeni objekat (kvazar) i određivanja položaja makro-likova, potrebno je uračunati uticaje kompaktnih objekata (zvezda) i međuzvezdane materije. U najvećem broju radova se za ceo sistem mikrosočiva usvaja da je *gladak*, zbog čega se za površinsku gustinu mase glatke galaksije uvodi parametar  $\kappa_s = \kappa_c + \kappa_*$ , gde su  $\kappa_*$  srednja površinska gustina mase kompaktnih objekata i  $\kappa_c$  površinska gustina mase međuzvezdane materije.

#### Primena numeričkog metoda u generisanju mapa gravitacionih sočiva

Ovim smo stigli do dela od najvećeg interesa za ovaj master rad, odnosno u nastavku opisujemo sistem gravitacionih mikrosočiva za koje računamo mapu pojačanja. Numerički metod, po kojem je realizovan program za račun mape gravitacionih mikrosočiva, se bazira na radu [29]. Pretpostavili smo da su sve zvezde u galaksiji jednake mase  $M$ , odnosno da su jednake mase Sunca, i da su slučajno raspoređene. Skala na kojoj opisujemo mikrosočiva u ravni sočiva se definiše preko Ajnštajnovog radijusa  $\xi_0$ , odnosno preko njegove projekcije na ravan izvora  $ERR = \xi_0 \frac{D_s}{D_l}$ . Takođe se uvodi lokalna gustina međuzvezdane materije  $\kappa_c$  izražena u jedinicama kritične površinske gustine mase  $\Sigma_{cr}$  ([29], [17]).

U programu koristimo **metod gađanja zracima** (inverzan *ray-tracing*) koji nam daje raspodelu pojačanja u ravni izvora usled efekta gravitacionih sočiva od strane skupa slučajno raspoređenih zvezda u ravni sočiva ([29], [18]). Ovakva raspodela pojačanja naziva se *mapa gravitacionih mikrosočiva*. Pod metodom gađanja zracima se podrazumeva da se prati svetlosni zrak koji se šalje iz položaja posmatrača na ravan sočiva. Zrak se u tom slučaju savija pod uticajem mikrosočiva u ravni sočiva, i pod tim uticajem pada na ravan izvora pod drugačijim uglom u odnosu na položaj koji bi zrak zauzeo u ravni izvora bez uticaja gravitacionih mikrosočiva.

Za jednačinu gravitacionih mikrosočiva je iskorišćena bezdimenziona jednačina u koordinatnom sistemu koji je rotiran tako da je paralelan sa spoljašnjim širom ([29], [17]):

$$\vec{y}(\vec{x}) = \begin{bmatrix} 1 - k_c - \gamma & 0 \\ 0 & 1 - k_c + \gamma \end{bmatrix} \vec{x} - \sum_{i=1}^N \frac{\vec{x} - \vec{x}_i}{|\vec{x} - \vec{x}_i|^2} \quad (4.1)$$

gde je sa  $\vec{y}$  opisan položaj izvora u ravni izvora, a sa  $\vec{x}$  prividni položaj izvora u ravni sočiva. Sa  $\vec{x}_i$  su definisani položaji slučajno raspoređenih zvezda (mikrosočiva) u ravni sočiva. Masa svake zvezde je izražena u jedinicama mase Sunca  $M_\odot$  i uvedena je pretpostavka da je jednaka masi Sunca.

Implementacija ovog numeričkog metoda može biti računski zahtevna ukoliko se za gravitaciona mikrosočiva usvoji veliki broj zvezda. Iz tog razloga se "shooting" oblast deli na manje oblasti, koje se u ravni sočiva ponašaju kao nezavisne celine što omogućuje daleko racionalniju potrošnju računarskog vremena.

### Algoritam

Algoritam je realizovan tako što su manje oblasti ravni gravitacionih mikrosočiva opisane matricom dimenzije  $2048 \times 2048$ . Za svaki element ove matrice se određuje vrednost potencijala na osnovu koje se dalje izračunava ugao savijanja zraka na kojeg bi dobijeni potencijal uticao. Iz relativnog položaja dotičnog elementa matrice i izračunatog ugla savijanja zraka na koji utiče ovaj element, u ravni izvora se određuje položaj savijenog zraka. Ovakvi zraci mogu pasti na isto mesto u ravni izvora, ili mogu pasti van oblasti koju smo definisali kao oblast proučavanja, odnosno van dimenzija mape. Na osnovu podataka o položajima savijenih zraka u ravni izvora, računa se broj zraka koji potom dodeljujemo mapi gravitacionih sočiva (opisanu promenjivom  $f_{\text{map}}$  u ovom radu).

Kako je pojačanje u svakom pikselu proporcionalno broju zraka ( $n_{\text{ray}}$ ) koji padaju na taj piksel, to će za matricu  $f$  važiti

$$f = 2.5 * \log_{10} \frac{f_{\text{map}}}{n_{\text{ray}}^2} \quad (4.2)$$

što se i koristi u implementaciji ovog algoritma.

Mapa gravitacionih mikrosočiva predstavlja matricu maksimalnih dimenzija  $6000 \times 6000$  piksela, koja korisniku daje vrednosti intenziteta pojačanja svakog izvora koji bi bio pokriven datim pikselom. Vrednosti pojačanja se računaju rešavanjem Poasonove jednačine.

---

## Rezultati paralelizacije i programske implementacije

---

### 5.1 Arhitektura klastera MISANU

Program za generisanje mapa gravitacionih mikrosočiva `ml.f` ([29], [16]), čije smo izvršavanje paralelizovali, nazvali smo `mlparallel.f`. Za paralelni računarski sistem je upotrebljen klaster "Matematički Institut - SANU" (MISANU) koji se sastoji od 17 nezavisnih čvorova, od kojih jedan na serveru IBM x3650 predstavlja *upravljajući čvor* (*eng.* management node), a ostalih 16 čvorova su IBM HS21 BladeCenter *računarski čvorovi* (*eng.* computation node) i oni zapravo čine HPC<sup>1</sup> klaster.

Za operativni sistem je izabrana Linux distribucija RedHat Enterprise Linux 5 za Intel x86\_64.

#### Korišćenje paralelnog programa

Za prevođenje i pokretanje MPI programa iskoristili smo MPICH prevodilac, koji predstavlja besplatnu verziju implementacije MPI standarda 1.2<sup>2</sup>. Program za generisanje mapa gravitacionih mikrosočiva je napisan u FORTRAN-u 77, zbog čega smo za prevođenje komandi iskoristili `pgf77` prevodilac. Za pokretanje programa koristi se naredba `mpirun` gde je glavni parametar broj procesa (procesora) na kojima će se program izvršavati, i ime programa koji se pokreće:

```
mpirun -np <broj procesa> <ime izvrsnog programa>
```

---

<sup>1</sup> *eng.* High Performance Computing

<sup>2</sup> MPI 1.2 sada predstavlja već staru veziju MPI biblioteke.

Komanda `mpirun` MPICH implementacije se koristi na različitim tipovima arhitektura klastera. Zbog ovoga komanda sama prepoznaje tip mašine na kojoj se poziva<sup>3</sup>. Opcija `-np` označava da se posle nje navodi broj procesa na koji će se paralelni program izvršavati. Kako se svaki čvor klastera MISANU sastoji od dva dvojez-garna procesora, broj procesorskih jedinica koji možemo istovremeno da iskoristimo je maksimalno jednak  $17 \times 2 \times 2$ , odnosno 68 procesorskih jedinica.

Zadavanje broja procesa radi izvršavanja paralelnog programa `mlparalel.f` zavisi od strukture petlje koju smo dekomponovali, zbog čega ćemo rezultate o optimalnom broju procesa predstaviti u rezultatima ovog rada.

### 5.1.1 Opis rada programa

Za postojeći sekvencijalni program (`ml.f`) razvili smo paralelnu verziju koda za generisanje mapa gravitacionih mikrosočiva. Izvršavanjem na klasteru MISANU, paralelni program dostiže ubrzanja od 10 do 40 puta, u zavisnosti od ulaznih parametara i broja zadatih procesa.

Ulazni podaci koji utiču na vreme izvršavanja paralelnog programa `mlparalel.f` su: dimenzije mape pojačanja (`npix`), konvergens ( $\kappa$ ), šir ( $\gamma$ ) i broj zraka kojima gađamo ravan sočiva (`nray`). Uzimajući u obzir da navedene četiri promenjive podjednako utiču na složenost našeg algoritma, praktično je nemoguće unapred odrediti na koji način bi promena svih ulaznih podataka mogla uticati na vreme izvršavanja. Iz ovog razloga smo ispitali uticaj promene svakog pojedinačnog parametra i rezultate predstavili u ovom radu.

Naša implementacija koristi *kolektivnu komunikaciju* (*eng.* collective communication) između procesora. Kolektivna komunikacija omogućuje grupi procesa da međusobno sarađuju u cilju distribucije i sakupljanja podataka među procesorima. Redukcija je jedna od ovakvih operacija i u našem programu `mlparalel.f` smo iskoristili funkciju `MPI_Reduce` koja podržava kolektivnu komunikaciju.

#### MPI\_Reduce

U cilju prikupljanja podataka nad kojima želimo da izvršimo operaciju redukcije, koristimo funkciju `MPI_Reduce`. Ova funkcija "sakuplja" podatke od procesa koji su deo komunikacione grupe `MPI_COMM_WORLD`. Funkcija dozvoljava izvršavanje osnovnih operacija nad podacima koji se šalju, kao što su sumiranje, određivanje maksimalnog broja, itd. Rezultat se sakuplja na procesoru čiji je identifikacioni broj naveden na mestu propisanom definicijom:

```
MPIREDUCE(
    sendbuf,          ! adresa prvog redukcionog elementa
```

<sup>3</sup> Kada to nije slučaj, potrebno je navesti tip mašine pomoću opcije `-machine`

```

    recvbuf ,          ! adresa prvog redukcionog rezultata
    INTEGER count ,    ! broj redukcija koje ce se izvrstiti
    INTEGER datatype , ! MPI tip elementa
    INTEGER operation , ! redukциони operator
    INTEGER root ,     ! procesor koji prima rezultat
    INTEGER comm       ! Komunikator
    INTEGER ierr )     ! FORTRAN argument za kontrolu greske

```

Parametar `sendbuf` predstavlja adresu promenjive čije se vrednosti šalju; `recvbuf` predstavlja adresu globalne promenjive u kojoj se smeštaju poslate vrednosti; `count` je ukupan broj elemenata koji jedan proces šalje; `datatype` predstavlja tip vrednosti promenjive `sendbuf`; `operation` vrstu operacije koju podržava funkcija `MPI_Reduce` i primenjuje se nad poslatim podacima; `root` je identifikacioni broj procesa na kojem se izvršava navedena operacija redukcije; `comm` je ime komunikatora, tj. grupa procesa koji učestvuju u redukciji.

Funkcija `MPI_Reduce` se izvršava na sledeći način: svaki proces poziva funkciju `MPI_Reduce`, bez čekanja na poziv `root`-a da započnu slanje. Podaci se prikupljaju na `root`-u, na kojem se onda izvršava operacija redukcije (`operation`), a rezultat se smešta u globalnu promenjivu `recvbuf`.

U nastavku predstavljamo pseudo-kod kojim smo opisali paralelizaciju sekvencijalnog programa `ml.f`. S obzirom na strukturu procesora na klasteru MISANU (dvojezgarni procesori), broj procesa na koje možemo da podelimo izvršavanje programa je veći od broja procesora (tj.  $> 32$ ). Iz ovih razloga umesto 'procesora' koristimo izraz 'procesi'.

## 5.2 Izvor paralelizacije

Program koji smo paralelizovali za izvršavanje na klasteru ima sledeću strukturu:

```

1  program mlparalel.f
2  include 'mpif.h'
3  {
4  ! Deklaracija promenljivih.
5  parametar maxpix
6  real f( maxpix , maxpix )
7
8  call MPI_Init;          ! Inicijalizacija MPI funkcija;
9  vreme_MPI = -MPI_Wtime(); ! Pocetak merenja MPI vremena
10 vreme_e = -etime();     ! Pocetak merenja Etime vremena
11 call MPI_Comm_Size; ! Odredjivanje broja procesa
12 call MPI_Comm_Rank; ! Dodeljivanje identifikacionog broja procesima
13
14 Odredjivanje položaja i ukupnog broja grav. mikrosociva (nlens);
15

```

```

16 Podela ravni sočiva na manje delove [m1 x m2] i odredjivanje njiho-
17 vih dimenzija preko ulaznih parametara konv i sir;
18
19   m = max (m1,m2);
20
21 ! Dekompozicija odgovarajuće petlje pozivom f-je para-range() u
22 ! zavisnosti od vrednosti za id procesa i m:
23   if ( m1 .eq. m ) -> dekompozicija sp. petlje na intervale
24                       [levi_m1 ,desni_m1]
25   else                -> dekompozicija un. petlje na intervale
26                       [levi_m2 ,desni_m2]
27
28   for ix = levi_m1; ix < desni_m1; ++ix
29     {
30     for iy = levi_m2, iy < desni_m2, ++iy
31       {
32       call potencijal(); ! Poziv funkcije potencijala koja
33                           ! racuna elemente matrice p;
34       Izracunavanje matrice f preko elemenata matrice p;
35       }
36     }
37
38   MPI_Reduce(f, f_map, MPLSUM); ! Sakupljanje i sabiranje matrica f
39                               ! i skladištenje rezultata u f_map
40   vreme_MPI = vreme_MPI + MPI_Wtime(); ! Kraj merenja MPI vremena
41   vreme_e = vreme_e + etime();        ! Kraj merenja Etime vremena
42
43   Ispis matrice f_map [npix,npix] u izlazni fajl;
44
45 }

```

Kao što smo naveli u opisu MPI implementacije, u samom zaglavlju je potrebno navesti ime biblioteke koja se poziva, gde je `mpif.h` poziv MPI biblioteke za FORTRAN (linija 2). Kako se matrice u FORTRAN77-u moraju definisati u zaglavlju programa (alociranje memorije nije omogućeno), uveden je parametar `maxpix` koji ograničava dimenzije matrica `f` i `fmap`<sup>4</sup> (linija 5). Na početku programa se svi elementi matrice `f[maxpix × maxpix]` inicijalizuju nulama, da bi se u nastavku operisalo samo nad `npix2` elemenata. Posle inicijalizacije MPI funkcija (linija 8), započinjemo računanje vremena rada paralelnog programa. Matrica `fmap` definiše mapu pojačanja gravitacionih mikrosočiva, koja se računa iz pomoćne matrice `f` (linija 38).

Na osnovu ulaznih parametara se određuje ukupan broj gravitacionih mikrosočiva u ravni sočiva (`nlens`) i određuju se broj ponavljanja petlji od koje će jedna biti dekomponovana (linije 14 i 16). Radi detaljnije analize paralelizacije opisaćemo na koji način smo došli na izloženu ideju paralelizacije.

<sup>4</sup> Parametar `maxpix` se može menjati u zavisnosti od vrednosti ulazne promenljive `npix`.

Izračunavanje elemenata matrice  $f(i,j)$ , od kojih zavise vrednosti elemenata  $f_{\text{map}}(i,j)$ , se realizuje unutar dve petlje glavnog dela programa. Petlje su definisane parametrima  $m_1$  i  $m_2$ , koji predstavljaju broj ponavljanja spoljašnje i unutrašnje petlje. Testiranjem sekvencijalnog koda je utvrđeno da se na izvršavanje ovog dela programa troši više od 98% ukupnog vremena, što nam je ukazalo na pogodne tačke paralelizacije.

Ispitivanjem vremena izvršavanja naredbi unutar ovih petlji zaključili smo da se najviše CPU vremena troši na poziv funkcije `potencijal`. Ova funkcija podrazumeva rešavanje Poasonove jednačine potencijala pomoću sinusnih Furijeovih transformacija (SFT)<sup>5</sup>. Ulazni parametri funkcije `potencijal` zavise od vrednosti ulaznih promenljivih programa: konvergenza i šira (tj.  $\kappa$  i  $\gamma$ ), iz kojih se računaju vrednosti elemenata matrice potencijala `p`, preko koje se dalje definiše matrica `f`. Prvobitno smo pokušali da izvršimo paralelizaciju same funkcije `potencijal` s obzirom da se SFT primenjuje nezavisno na vrstama i kolonama matrice `p`. Paralelizacija se bazirala na podeli matrice `p` na blokove koji će se jednako distribuirati među procesima. Svaki proces je računao elemente svoje podmatrice i rezultat slao master procesu, koji bi dobijene elemente sjedinio u jednu matricu. Ovo bi se onda ponovilo još 3 puta, primenom SFT-a na kolone podmatrica, pa potom opet na vrstama, i četvrti put opet nad kolonama. Eksperimentalno je utvrđeno da ovo ne daje pogodne rezultate i da se ubrzanja postižu tek za veće "dimenzije" problema<sup>6</sup> u kom slučaju brzina operacija nadmašuje brzinu komunikacije među procesorima. Ovo je očigledno uticalo da se drugačije pristupi paralelizaciji postojećeg sekvencijalnog programa `ml.f`.

Ideja koja nam je omogućila da dobijemo dobra ubrzanja se svodi na podelu unutrašnje ili spoljašnje petlje (zavisno od maksimalne vrednosti  $m_1$  i  $m_2$ ) među procesima, odnosno na dekompoziciju jedne od petlji tako što svakom procesu dodeljujemo različite intervale za koje se računaju elementi matrice `f`. Kako  $m_1$  i  $m_2$  mogu imati različite vrednosti (u zavisnosti od ulaznih podataka), radi boljeg iskorišćenja resursa klastera<sup>7</sup> definisali smo proceduru `para_range` koja na osnovu maksimuma vrednosti  $m_1$  i  $m_2$  utvrđuje način na koji će se izvršiti dekompozicija tj. da li će to biti spoljašnja ili unutrašnja petlja, i opseg intervala u sklopu kojih se računaju elementi matrice `f` na svakom procesu. Ovo nam je obezbedilo ravnomernu podelu posla među procesima. Po završetku rada spoljašnje petlje, izvršava se redukcija. Redukcija se obavlja tako što svaki proces šalje svoju verziju matrice `f`, koje se potom sabiraju u jednu jedinstvenu matricu  $f_{\text{map}}$  pomoću MPI funkcije `MPI_Reduce`.

---

<sup>5</sup> Brze Furijeove transformacije (FFT) nisu korišćene zbog dimenzija matrice nad kojom se primenjuju transformacije (FT) i koje ne moraju biti jednake stepenu dvojke  $2^n$ . Primena FFT-a (*eng.* Fast Fourier Transformation) na  $n \neq 2^n$  može da prouzrokuje neželjene greške u radu sa brojevima sa dvostrukom preciznošću.

<sup>6</sup> Dimenzije se povećavaju sa povećavanjem vrednosti ulaznih parametara.

<sup>7</sup> U ovom slučaju se misli na klastere sa proizvoljnim brojem čvorova.



## 5.3 Rezultati

### 5.3.1 Analiza vremena izvršavanja paralelnog programa

Za određivanje vremena rada paralelnog programa iskoristili smo dve procedure: `etime` i `MPI_Wtime`. `Etime` vraća realan broj (`REAL(4)`) koji predstavlja proteklo vreme mereno od nekog proizvoljnog trenutka (*eng.* elapsed time). `Mpi_Wtime` je deo MPI standarda i ova procedura vraća šezdesetčetvorobitnu realnu vrednost koja predstavlja proteklo vreme časovnika mereno u sekundama u odnosu na neki proizvoljan trenutak u prošlosti. `MPI_Wtime` daje lokalni rezultat, odnosno vezano je za procesor na kojem se izvršava.

`Etime` isključivo meri vreme izvršavanja sekvencijalnih naredbi paralelnog programa, tj. ne uključujući naredbe komunikacije između procesa. S druge strane, `MPI_Wtime` meri i vreme komunikacije, odnosno u našem slučaju, vreme potrebno da proces pošalje ili primi svoje podatke. Prilikom komunikacije između procesa može doći do privremenog zastoja, gde jedan proces čeka da drugi proces završi sa slanjem svojih podataka. U tom slučaju je za analizu vremena izvršavanja potrebno uključiti i vreme čekanja (*eng.* idle time) karakteristično za situacije u kojima više procesa istovremeno šalje svoje rezultate. Ovo se odražava i u našem paralelnom programu s obzirom da svi procesi šalju vrednosti matrice `f` ka master procesu (označen kao `root`). Jedini način da se približno ustanovi vreme komunikacije između više procesa jeste preko razlike `Etime` i `MPI_Wtime` vremena. Međutim, zbog jednog poziva redukcije (tj. samo jednog poziva na komunikaciju) i velikog variranja u dobijenim rezultatima za oba vremena, nismo u mogućnosti da tačno odgovorimo na pitanje komunikacije između različitih procesorskih elemenata klastera (čvorovima, procesorima i procesorskim jezgrima).

Postoji izvesna složena zavisnost između nekih ulaznih parametara i broja ponavljanja ciklusa kojeg smo dekomponovali. Takva je situacija u slučaju konvergensa i šira. Da bismo predvideli vreme izvršavanja potrebno je pojedinačno analizirati uticaj svakog ulaznog parametra (`npix`,  $\kappa$ ,  $\gamma$  i `nray`).

#### Konvergens i šir

Konvergenes ( $\kappa$ ) i šir ( $\gamma$ ) utiču na vreme izvršavanja programa preko komplikovane formule u kojoj se istovremeno računa koren i razlika ova dva parametara, zbog čega je teško predvideti vreme izvršavanja paralelnog programa u zavisnosti od promena njihovih vrednosti.

Program `mlparalel.f` smo testirali za  $n = 2^l, l \in \{1, 2, 3, 4, 5, 6\}$  (gde je  $n$  broj zadatih procesa) za koje smo utvrdili vremena izvršavanja našeg paralelnog programa. U tabeli 5.1 su predstavljeni rezultati testa za različite vrednosti ulaznog parametra konvergensa ( $\kappa$ ) i odgovarajućeg broja mikrosočiva (`nlens`).

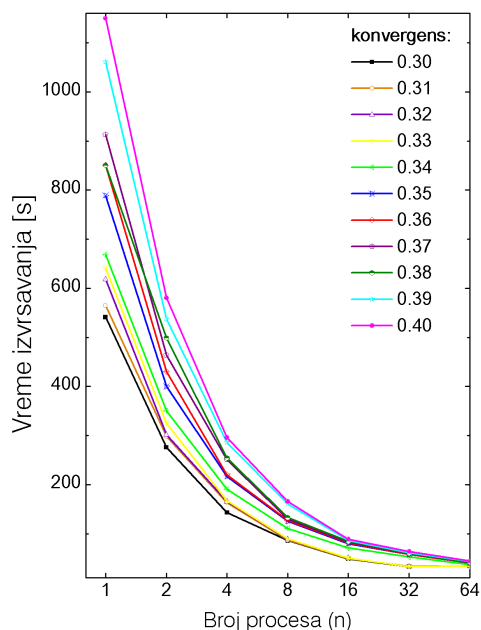
Tabela 5.1: Vremena izvršavanja paralelnog programa (izražena u sekundama) na različitom broju procesa ( $n$ ), za različite vrednosti konvergensa ( $\kappa$ ) i broja mikrosočiva ( $nlens$ ).

$\kappa$	$nlens$	Vreme [s]						
		$n=1$	$n=2$	$n=4$	$n=8$	$n=16$	$n=32$	$n=64$
0.3	940	541.21	275.95	143.47	86.16	49.19	33.13	34
0.31	1048	564.81	298.14	163.68	86.90	49.57	33.80	34.18
0.32	1178	618.29	301.79	165.97	88.63	50.33	33.39	34.34
0.33	1317	640.53	325.38	167.57	89.59	50.65	33.30	34.7
0.34	1465	668.01	350.24	190.43	110.57	71.38	52.57	36.98
0.35	1652	788.96	399.81	216.67	125.21	80.20	58.34	40.17
0.36	1866	850.52	430.39	220.57	127.23	81.34	58.23	40.81
0.37	2114	913.31	463.85	251.25	130.06	83.27	59.30	40.72
0.38	2397	850.52	499.05	254.16	132.69	84.51	60.35	41.16
0.39	2713	1060.92	537.24	286.12	160.79	86.52	61.48	43.36
0.4	3104	1149.92	580.55	295.87	165.95	89.10	64.12	44.12

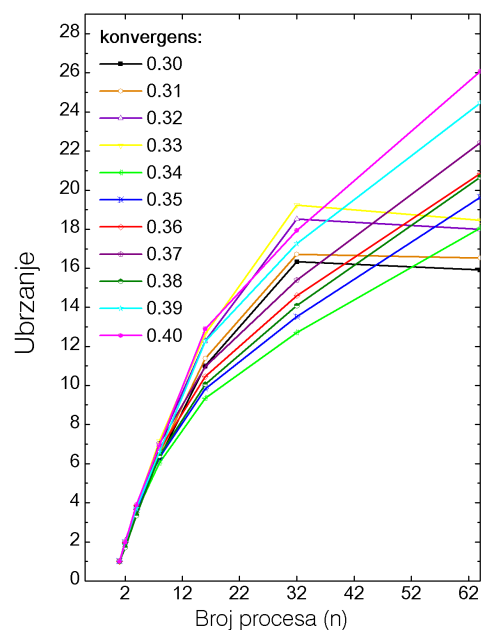
Testiranje smo izvršili tako što smo tri ulazna parametra fiksirali na  $npix = 1000$ ,  $\gamma = 0.4$ ,  $nray = 10$ , sa ciljem da utvrdimo na koji način će se vreme rada paralelnog programa menjati za različite vrednosti konvergensa. Parametar  $\kappa$  smo zato varirali od 0.3 do 0.4, sa korakom  $\Delta\kappa = 0.01$ . Za svako  $\kappa$  smo izvršili vremensku analizu puštanjem paralelnog programa na različitom broju procesa  $n = 2^l, l \in \{1, 2, 3, 4, 5, 6\}$ . Ovo smo ponovili dva puta za svako  $n$ , te su u tabeli 5.1 izložene srednje vrednosti vremena izvršavanja, koje ćemo ovde označiti sa  $t(\kappa, n)$ . Odstupanja od ovih vrednosti su u granicama  $\pm 2^s$ , što je zanemarljivo za utvrđivanje ubrzanja paralelnog programa.

Na grafiku 5.1 su, za različite vrednosti konvergensa, predstavljene krive  $t(\kappa, n)$ , definisane na osnovu rezultata iz tabele 5.1. Kao što se može primetiti, za veće vrednosti  $\kappa$  izvršavanje paralelnog programa na jednom procesu ( $t(\kappa, 1)$ ) duže traje. Ukoliko program pustimo na većem broju procesa, vreme  $t(\kappa, n)$  za svako fiksirano  $\kappa$  znatno opada, tj. paralelni program se sve brže izvršava.

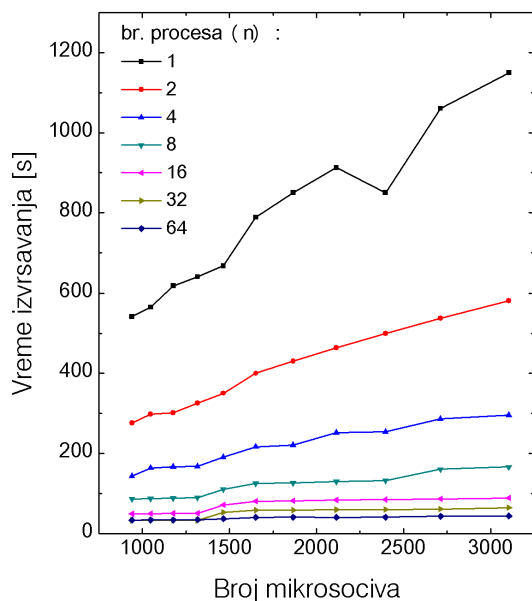
Na grafiku 5.2 smo prikazali *ubrzanja* izvršavanja paralelnog programa zavisno od zadatog broja procesa  $n$ . Ubrzanja koja se postižu za  $\kappa < 0.33$  u jednom trenutku prestaju sa rastom, odnosno ove četiri krive dostižu maksimum za  $n = 34$ . Ovo je zato što je broj ponavljanja petlje  $m$  (koju dekomponujemo) ograničen vrednostima konvergensa,. Tačnije, za  $\kappa < 0.33$  će  $m$  imati vrednosti ispod 34, te ukoliko za broj procesa zadamo broj  $> 34$  aktiviraju se "prazni" procesi, koji ne doprinose paralelizaciji jer ne pozivaju funkciju `potencijal`, ali učestvuju u ukupnom radu paralelnog programa te ga samo vremenski opterećuju. Kod ostalih krivih ubrzanja vidimo da dolazi do rasta. To se dešava jer  $m$  dobija vrednosti veće od 34, zbog čega se za  $n > 34$  smanjuje broj praznih procesa. Zato je radi smanjivanja vremena izvršavanja programa bitno dobro proceniti vrednost za  $n$ , a ovde smo pokrili samo neke od mogućih slučajeva.



Slika 5.1: Vreme izvršavanja programa postignuto na  $2^l$  čvorova klastera MISANU za različite vrednosti **konvergensa**.



Slika 5.2: Ubrzanje paralelnog programa postignuto na  $2^l$  čvorova klastera MISANU za različite vrednosti **konvergensa**.



Slika 5.3: Grafik vremena izvršavanja programa u zavisnosti od manjeg **broja mikrosočiva**.

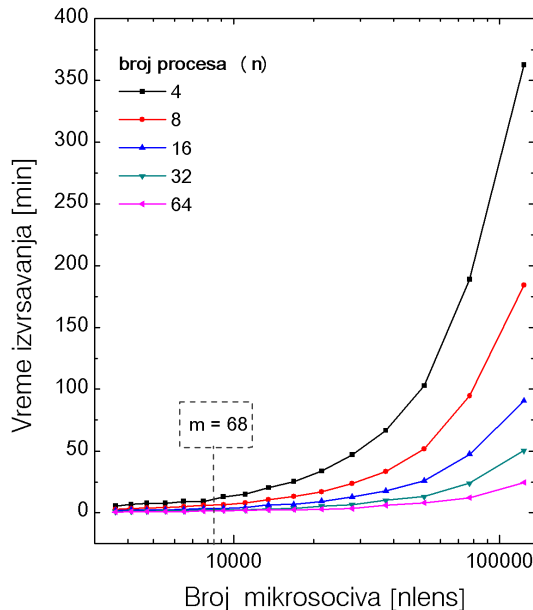
Ilustrativniji prikaz uticaja promene konvergensa na performanse paralelnog izvršavanja je dat grafikom 5.3, gde smo kao osnovni parametar iskoristili broj mikrosočiva (nlens) koji, takoreći, materijalizuje parametar  $\kappa$ .

Na grafiku 5.3 smo, na osnovu tabele 5.1, predstavili vremenske krive izvršavanja paralelnog programa u zavisnosti od promene broja mikrosočiva u ravni sočiva. Sa grafika vidimo da vremenska kriva za slučaj  $n=1$  raste zajedno sa rastom broja mikrosočiva, što je i jasno usled povećanja broja izračunavanja. Ono što je interesantno jeste slučaj za  $n=64$  gde imamo skoro konstantan trend. Kako bi ispitali ponašanje rada programa za veći broj mikrosočiva, u tabeli 5.2 smo izložili vreme izvršavanja programa za vrednosti konvergensa  $0.41 \leq \text{nlens} \leq 0.56$ .

Tabela 5.2: Vremena izvršavanja paralelnog programa (izražena u minutima) na različitom broju procesa ( $n$ ), za različite vrednosti konvergensa ( $\kappa$ ) i broja mikrosočiva ( $n_{lens}$ ).

$\kappa$	$n_{lens}$	Vreme [min]				
		$n=4$	$n=8$	$n=16$	$n=32$	$n=64$
0.41	3597	5.48	2.77	1.53	1.09	0.74
0.42	4118	6.83	3.72	2.26	1.27	0.82
0.43	4722	7.67	3.88	2.35	1.31	0.87
0.44	5541	7.68	4.10	2.48	1.37	0.89
0.45	6504	9.22	4.97	2.62	1.43	0.94
0.46	7708	9.21	5.99	3.46	2.17	1.51
0.47	9163	12.86	6.47	3.75	2.32	1.63
0.48	11065	14.92	7.90	4.09	2.54	1.77
0.49	13549	20.32	10.67	6.45	3.12	2.19
0.5	16856	25.41	13.29	6.79	3.55	2.45
0.51	21463	33.70	16.88	9.23	5.36	2.87
0.52	27906	46.88	23.50	12.67	6.46	3.42
0.53	37407	66.53	33.33	17.78	9.99	6.07
0.54	52167	102.89	51.53	25.89	13.07	8.06
0.55	77171	188.93	94.59	47.43	23.85	12.12
0.56	123745	362.79	184.20	90.73	50.29	24.48

Na grafiku 5.4 se nalaze krive koje predstavljaju promenu vremena u zavisnosti od promene broja mikrosočiva, na osnovu rezultata iz tabele 5.2.



Slika 5.4: Grafik vremena izvršavanja programa u zavisnosti od većeg broja mikrosočiva.

nam ukazuju na dobru raspoređenost paralelnih zadataka među više procesa. Za broj mikrosočiva veći od 8000, broj  $m$  počinje naglo da raste i svaki proces će znatno više puta da poziva funkciju potencijala.

Svaku vrednost konvergensa (tj. odgovarajućeg broja mikrosočiva) karakteriše vrednost parametra  $m$  (broj ponavljanja glavne petlje). Ispitujući ponašanje parametra  $m$  u zavisnosti od promene konvergensa, primetili smo da vreme izvršavanja započinje znatan rast onda kada broj ponavljanja petlje koja se dekomponuje, postane veće od 68. Kao što smo objasnili na strani 67, aktivnost procesa zavisi od broja  $m$ , te ukoliko je taj broj veći ili jednak maksimalnom broju procesorskih jedinica sa kojim jedan klaster raspolaže, na jednom procesu interval [levi, desni] raste, zbog čega rastu i pozivi funkcije `potencijal` znatno više nego što je to u slučaju  $m < 68$ . Ovu situaciju smo označili isprekidanom linijom na grafiku 5.4. Delovi krivih pre isprekidane linije

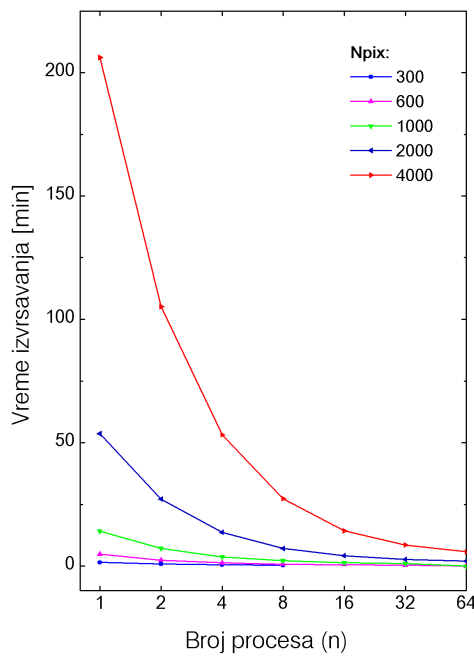
## Broj piksela mape i zraka svetlosti

Za razliku od prethodnih ulaznih parametara, za promenjive  $npix$  i  $nray$  se može izvesti algoritamska složenost. Na osnovu formula kojima je opisana zavisnost broja ponavljanja ciklusa ( $m1$  i  $m2$ ) i broja piksela, može se reći da u slučaju dimenzije mape pojačanja ( $npix$ ) algoritamska složenost je  $\mathcal{O}(npix^2)$ . Pod ovim se podrazumeva da su ostali ulazni parametri tokom vremenske analize fiksirani. Isti zaključak se izvodi u slučaju broja zraka, te za ostale fiksirane parametre će važiti da se složenost algoritma može opisati preko  $\mathcal{O}(nray^2)$ .

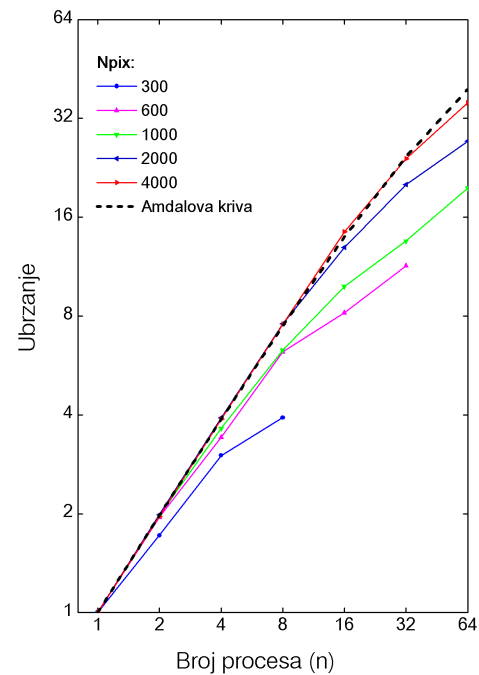
Paralelni program smo testirali za ulazne vrednosti:  $\gamma = 0.4$ ,  $nray = 10$ , pri čemu smo za konvergenciju izabrali vrednosti 0.36, a rezultate predstavili u tabeli 5.3.

Tabela 5.3: Vremena izvršavanja paralelnog programa (izražena u minutima) na različitom broju procesa ( $n$ ), za različite vrednosti dimenzije mape pojačanja ( $npix$ ).

npix	Vreme[min]						
	n=1	n=2	n=4	n=8	n=16	n=32	n=64
300	1.44	0.84	0.48	0.37	-	-	-
600	4.82	2.47	1.41	0.77	0.59	0.42	0.00
1000	14.13	7.18	3.68	2.13	1.39	1.03	0.00
2000	53.67	27.07	13.68	7.08	4.15	2.67	1.97
4000	206.17	105.15	53.17	27.28	14.27	8.55	5.77



Slika 5.5: Vreme izvršavanja programa postignuto na  $2^l$  čvorova klastera MISANU za različite vrednosti  $npix$ .



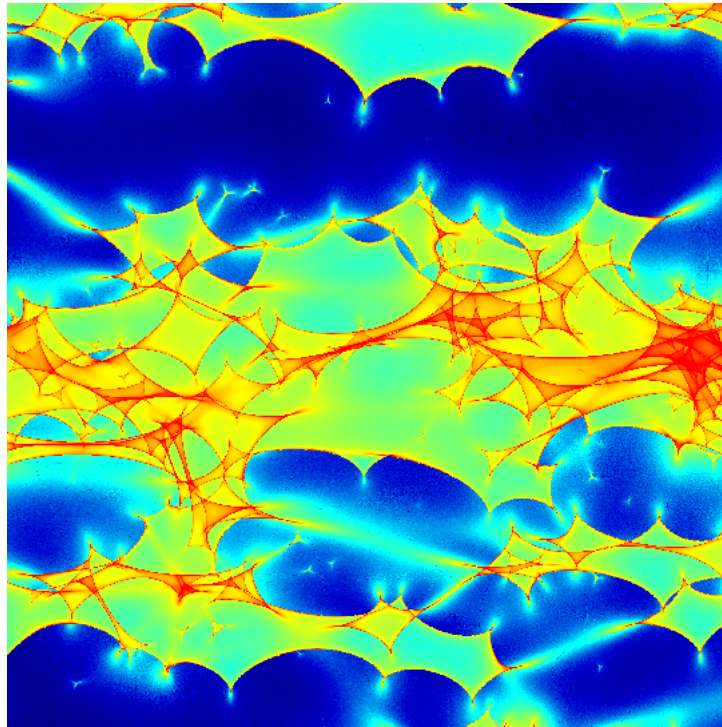
Slika 5.6: Ubrzanje paralelnog programa postignuto na  $2^l$  čvorova klastera MISANU za različite vrednosti  $npix$ .

Na grafiku 5.5 je predstavljena kriva zavisnosti vremena izvršavanja programa na različitom broju procesa. Kriva za  $\text{npix}=4000$  je najdominantija i za ove vrednosti dimenzija je potrebno najviše vremena da bi se izračunala mapa pojačanja.

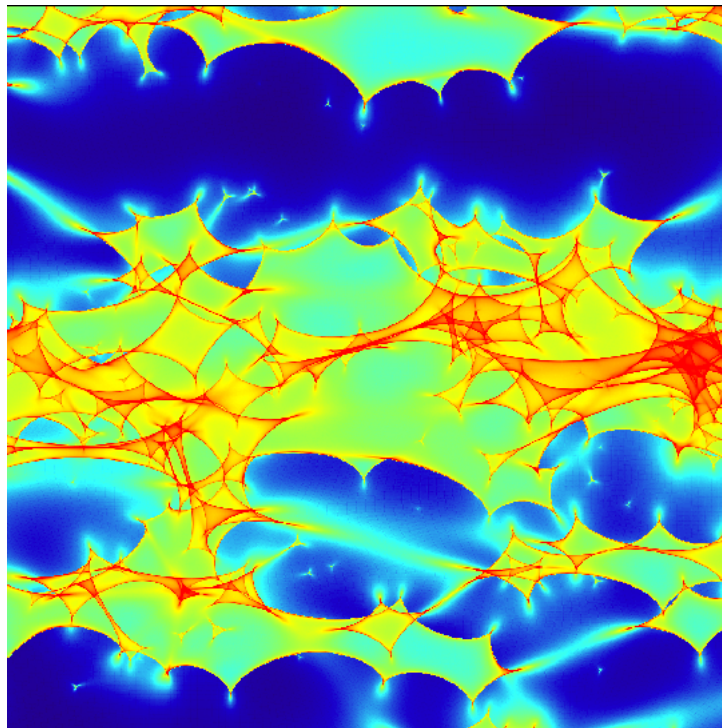
Na grafiku 5.6 smo isprekidanom linijom predstavili maksimalno ubrzanje ukoliko bi se program, u kojem je 99% naredbi paralelizovano, izvršavao bez uticaja komunikacije (izvedeno iz Amdalovog zakona). Na istom grafiku su prikazane empirijske krive ubrzanja definisane na osnovu vrednosti tabele 5.3. Vidimo da se sa povećanjem vrednosti za  $\text{npix}$  povećava i ubrzanje programa, odnosno resursi klastera se sve više koriste.

Na osnovu vremena rada programa `m1paralel.f` za prvih šest slučajeva iz tabele 5.2 i na osnovu krive za  $n = 64$ , izvodimo zaključak da komunikacija među procesima ne utiče značajno na brzo izvršavanje paralelnog programa `m1paralel.f` i da se sa dupliranjem broja procesorskih jedinica ubrzanje takođe duplira.

Najbitniji rezultat ovog rada jeste mogućnost generisanja mapa pojačanja za velike vrednosti broja zraka ( $\text{nray}$ ), što zahteva puno vremena u slučaju rada sekvencijalnog programa `m1.f` na standardnim PC računarima. Program smo testirali na kvazaru QSO 2237+0305A, za  $\kappa = 0.36$ ,  $\gamma = 0.4$  i  $\text{nray} = 10$ . Dobijena mapa pojačanja predstavlja vrednosti intenziteta pojačanja zraka svetlosti koji prolazi kroz sistem slučajno raspoređenih zvezda. Vreme potrebno za generisanje mape dimenzija  $2000 \times 2000$  i za ulazne parametre QSO 2237+0305A, na jednom procesoru klastera MISANU iznosi jedan sat. Ubrzanja koja se postižu ukoliko iskoristimo 16, odnosno 64 procesa, su 13 odnosno 30 puta. Priloženu sliku 5.7 na 64 procesa (tj. jezgara) generisali smo za dva minuta. Druga priložena slika 5.8 predstavlja mapu pojačanja dimenzija  $1000 \times 1000$  za  $\text{nray} = 256$ , koju smo izračunali za  $2^h 15^{min}$ . Kao što se može primetiti, bez obzira na manju dimenziju mape druga slika 5.8 daje jasnije detalje i time preciznije informacije o uticaju modela sistema gravitacionih mikrosočiva na zrake svetlosti.



Slika 5.7: Mapa uvećanja kvazara QSO 2237+0305A dobijena za  $n_{ray} = 10$ .



Slika 5.8: Mapa uvećanja kvazara QSO 2237+0305A dobijena za  $n_{ray} = 256$ .

### 5.3.2 Broj procesa. Karakteristike klastera

Zahvaljujući testiranju paralelnog programa `mlparallel.f` stekli smo uvid u ponašanje procedura i komandi iz MPICH implementacije na klasteru MISANU. Saznali smo na koji način se vrši distribucija zadataka među čvorovima (procesorima) klastera, ograničenja koja su propisana MPI standardom za veličinu podataka koji se razmenjuju među procesorima i koji je optimalan broj procesa za koji dobijamo najbolje vreme izvršavanja paralelnog programa.

Prvi rezultat koji ovde iznosimo jeste pravilo "zakazivanja" (*eng.* scheduling) zadataka na procesorskim elementima klastera. Utvrdili smo da MPICH rezerviše procese po čvorovima, odnosno traži prvi slobodan procesor sa svakog čvora. Ukoliko je zadati broj procesa veći od postojećeg broja čvorova na klasteru, MPICH se vraća na početak liste čvorova i započinje zakazivanje njihovog drugog procesora, ukoliko je procesor slobodan. Ovo lepo ilustruje situaciju kada se za broj procesa zadaju vrednosti manje od 34. Ukoliko je broj procesa veći od 34, procesi se dodeljuju jezgrima procesora. Pod "dodeljivanjem" smatramo distribuciju izvršnog programa, gde će svaki proces imati svoju kopiju.

Drugi rezultat koji smo dobili je vezan za ponašanje MPICH-a prilikom distribucije paralelnog programa na veći broj procesa nego što je procesorskih jedinica (jezgara) na klasteru. Ovakva situacija je poznata pod nazivom *opterećenje čvorova* (*eng.* oversubscribing nodes) iz razloga što MPICH svim procesima daje jednaki prioritet pri izvršavanju. U tom slučaju procesi, kojih sada ima više na jednom jezgru, će se "boriti" da izvrše svoje kopije programa što može dovesti do značajne degradacije u performansama programa. Ipak, postoje situacije kada je vremenski isplativije koristiti veći broj procesa, bez obzira na pad iskorišćenosti procesorskih jedinica.

Ovde stižemo do trećeg rezultata koji se odnosi na optimalan broj procesa koji zadajemo kao argument komande `mpirun`. Da bismo ga odredili, razvili smo program koji nam na standardni izlaz izbacuje vrednost broja ponavljanja ciklusa koji će biti dekomponovan prilikom izvršavanja paralelnog programa. Utvrdili smo da najbolje vreme izvršavanja paralelnog programa dobijamo ukoliko je broj procesa jednak broju ovih ponavljanja, čak i situacijama opterećenja čvorova. Bitno je naglasiti da postoji i gornje ograničenje na broj zadatah procesa. S obzirom da ovaj broj nismo znali unapred (nije naveden u MPICH pomoćnim tekstovima jer zavisi od karakteristika klastera), testiranjem smo utvrdili da iznosi **170**. Ukoliko za broj procesa zadamo vrednost veću od 170, MPI i operativni sistem prekidaju izvršavanje paralelnog programa, izbacujući na standardni izlaz obaveštenje da je prilikom komunikacije procesa došlo do greške.

Poslednji rezultat koji je interesantan sa stanovišta upotrebe programa `mlparallel.f` jeste ograničenje dimenzija pomoćne matrice `f`, koje se pri redukciji šalju na master čvor. Testovima smo utvrdili maksimalnu vrednost za `maxpix` u zavisnosti od



tipa vrednosti elemenata matrice  $f$ . U slučaju realnog broja dvostruke preciznosti (`double precision`), najveća vrednost koju možemo zadati za `maxpix` iznosi **5729**, dok u slučaju jednostruke preciznosti (`real`) najveća vrednost iznosi **11584**. Uzrok ovome je upravo u MPICH implementaciji primenom ograničenja na veličinu podataka koji se šalju preko 1Gb međuprocesorske mreže.

## 5.4 Zaključak

U ovom radu su predstavljeni rezultati paralelizacije sekvencijalnog programa za generisanje mapa pojačanja gravitacionih mikrosočiva. Analiza vremena izvršavanja paralelnog programa je pokazala da to vreme u velikoj meri zavisi od parametara gravitacionih mikrosočiva (`kovergens` i `šir`), kao i od dimenzija mape (`npix`) i broja zraka po pikselu (`nray`). Vreme izvršavanja se znatno smanjuje, a odgovarajuće ubrzanje znatno povećava, sa povećanjem broja procesa. To omogućuje da se generišu mape pojačanja gravitacionih mikrosočiva i za one parametre za koje bi odgovarajuće vreme izvršavanja sekvencijalnog programa trajalo toliko dugo da bi onemogućilo njegovu praktičnu primenu. Zbog toga rezultati dobijeni u ovoj tezi, kao i sam paralelizovani program, imaju veliki značaj za praktičnu astronomsku primenu. Osim toga, testiranje paralelizovanog programa na klasteru MISANU je omogućila da se stekne uvid u optimalan broj procesa za koje se dobija najbolje vreme izvršavanja, kao i nekih drugih aktuelnih karakteristika klastera MISANU.

---

## Reference

---

- [1] G. Amdahl. Validity of the single processor approach to achieving large scale computing capabilities. *AFIPS Conference Proceedings*, **30**:483–485, April 1967.
- [2] K. Chang and S. Refsdal. Flux variations of QSO 0957+561 A, B and image splitting by stars near the light path. *Nature*, **282**:561, 1979.
- [3] K. Chang and S. Refsdal. Star disturbances in gravitational lens galaxies. *Astronomy and Astrophysics*, **132**:168, 1984.
- [4] D. Clowe, M. Bradač, A. H. Gonzalez, and et al. A Direct Empirical Proof of the Existence of Dark Matter. *The Astrophysical Journal*, **648**:L109–L113, 2006.
- [5] Alexander Eigenbrod. *Astrophysical Applications of Gravitationally Lensed Quasars: From Dark Matter Halos to the Structure of Quasars Accretion Disks*. PhD thesis, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland, 2008.
- [6] Albert Einstein. Die Grundlage der allgemeinen Relativitätstheorie. *Annalen der Physik*, **49**:769, 1916.
- [7] C. Alcock et al. Possible gravitational microlensing of a star in the Large Magellanic Cloud. *Nature*, **365**:621, August 1993.
- [8] E. Aubourg et al. Evidence for gravitational microlensing by dark objects in the Galactic halo. *Nature*, **365**:623, 1993.
- [9] Ian W. Browne et al. The Cosmic Lens All-Sky Survey - II. Gravitational lens candidate selection and follow-up. *Monthly Notices of the Royal Astronomical Society*, **341**:13, 2003.
- [10] J. P. Beaulieu et al. Discovery of a cool planet of 5.5 Earth masses through gravitational microlensing. *Nature*, **439**:437, 2006.

- 
- [11] Van Waerbeke et al. Detection of correlated galaxy ellipticities from CFHT data: first evidence for gravitational lensing by large-scale structures. *Nature*, **358**:30, 2000.
- [12] Vladimir Filipović. *Operatori selekcije i migracije i Web servisi kod paralelnih evolutivnih algoritama*. PhD thesis, Matematički Fakultet Univerziteta u Beogradu, 2006.
- [13] J. N. Hewitt, E. L. Turner, D. P. Schneider, B. F. Burke, and G. I. Langston. Unusual radio source MG1131+0456 - A possible Einstein ring. *Nature*, **333**:537, 1988.
- [14] M. J. Irwin. Photometric variations in the Q2237 + 0305 system - First detection of a microlensing event. *Astronomical Journal*, **98**:1989–1994, 1989.
- [15] Newton Isaac. *The Third Book of Optick*, 1717.
- [16] P. Jovanović, A. F. Zakharov, L. Č. Popović, and T. Petrović. Microlensing of the X-ray, UV and optical emission regions of quasars: simulations of the time-scales and amplitude variations of microlensing events. *Monthly Notices of the Royal Astronomical Society*, **386**:397, 2008.
- [17] Predrag Jovanović. *Uticaj gravitacionih sočiva na X-zračenje akrecionog diska kod aktivnih galaksija*. PhD thesis, Matematički fakultet Univerziteta u Beogradu, Beograd, 2005.
- [18] R. Kayser, S. Refsdal, and R. Stabell. Astrophysical applications of gravitational micro-lensing. *Astronomy and Astrophysics*, **166**:36–52, 1986.
- [19] Lindsay J. King, Ian W. Browne, Daniel R. Marlow, Alok R. Patnaik, and Peter N. Wilkinson. Gravitationally lensed radio sources in the Jodrell Bank-VLA Astrometric Survey. *Monthly Notices of the Royal Astronomical Society*, **307**:225, 1999.
- [20] Jozef J. Kratica. *Paralelizacija funkcionalnih programskih jezika i implementacija na transpjuterskim sistemima*. Master's thesis, Matematički Fakultet Univerziteta u Beogradu, 1994.
- [21] Jozef J. Kratica. *Paralelizacija genetskih algoritama za rešavanje nekih NP-kompletnih problema*. PhD thesis, Matematički Fakultet Univerziteta u Beogradu, 2000.
- [22] D. Maoz, J. N. Bahcall, and D. P. Schneider et al. The Hubble Space Telescope Snapshot Survey. IV - A summary of the search for gravitationally lensed quasars. *Astrophysical Journal*, **409**:28, 1993.
- [23] Massimo Meneghetti. *Introduction to Gravitational Lensing. Lecture scripts*, 2008.

- 
- [24] Ramesh Narayan and Matthias Bartelmann. Lectures on Gravitational Lensing. *eprint arXiv:astro-ph/9606001*, 1996.
- [25] Bohdan Paczyński. Gravitational Microlensing at Large Optical Depth. *The Astrophysical Journal*, **301**:503–516, 1986.
- [26] Bohdan Paczyński. Gravitational microlensing by the galactic halo. *The Astrophysical Journal*, **304**:1–5, 1986.
- [27] Bohdan Paczyński. Giant luminous arcs discovered in two clusters of galaxies. *Nature*, **325**:572, 1987.
- [28] Michael J. Quinn. *Parallel Programming in C with MPI and OpenMP*. McGraw-Hill Companies, New York, 2003.
- [29] P. Schneider and A. Weiss. A gravitational lens origin for AGN-variability? Consequences of micro-lensing. *Astronomy and Astrophysics*, **171**:49–65, 1987.
- [30] Peter Schneider, Christopher S. Kochanek, and Joachim Wambsganss. *Gravitational Lensing: Strong, Weak and Micro*. Springer, Berlin, Heidelberg, 2006.
- [31] Bernard Schutz. *A First Course in General Relativity*. Cambridge, Max Planck Institute for Gravitational Physics, 2009.
- [32] A. Stockton. The lens galaxy of the twin QSO 0957+561. *Astrophysical Journal*, **242**:L141, 1980.
- [33] David W. Walker. Standards for message-passing in a distributed memory environment. Technical report, Oak Ridge National Laboratory, August, 1992. First CRPC Workshop on Standards for Message Passing in a Distributed Memory Environment, April 1992.
- [34] D. Walsh, R. F. Carswell, and R. J. Weymann. 0957 + 561 A, B - Twin quasar objects or gravitational lens. *Nature*, **279**:381–384, 1979.
- [35] Joachim Wambsganss. Cosmological Microlensing. In Sackett Penny D. Menzies J. W., editor, *Microlensing 2000: A New Era of Microlensing Astrophysics*, volume **239** of *ASP Conference Proceedings*, page 351, 2001.
- [36] Joachim Wambsganss. Microlensing Search for Dark Matter at all Mass Scales. In Yannick Mellier, editor, *Impact of Gravitational Lensing on Cosmology*, Proceedings of IAU Symposium, 2004.
- [37] Peter Young. The double quasar Q0957 + 561 A, B - A gravitational lens image formed by a galaxy at  $Z = 0.39$ . *Astrophysical Journal*, **241**:507, 1980.
- [38] Peter Young. Q0957+561: Effects of Random Stars on the Gravitational Lens. *Astrophysical Journal*, **244**:756, 1981.

- 
- [39] Fritz Zwicky. Nebulae as Gravitational Lenses. *Physical Review*, **51**:290, 1937.
- [40] Fritz Zwicky. On the Probability of Detecting Nebulae Which Act as Gravitational Lenses. *Physical Review*, **51**:679, 1937.

## A.1 Kratka istorija razvoja paralelnih računara

Za početak razvoja pravog (MIMD) paralelizma zaslužni su Federiko Luidi (Federico Luigi), Konte Menebre (Conte Menebrea) sa radom iz 1942. godine [L. F. Menebrea, 1942]. Godine 1954. IBM je proizveo 704 (FORTRAN i Lisp su razvijeni za 704), zahvaljujući učešću programera Žene Amdala (*eng.* Gene Amdahl) koji je bio jedan od vodećih arhitekata. IBM 704 je postao prvi komercijalno dostupan računar koji je koristio punu automatizaciju komandi nad aritmetikom u pokretnom zarezu, sa kapacitetom od 5kFLOPS-a. U aprilu 1958. godine Gil (S. Gill Ferrati) je govorio o paralelnom programiranju. Iste godine IBM-ovi istraživači Džon Kok (John Cocke) i Danijel Slotnik (Daniel Slotnik) su diskutovali o korišćenju paralelizma za numeričke proračune. Borou (Borroughs) korporacija je 1962. godine uvela D825, 4-procesorski računar sa pristupom ka 16 memorijskih modula kroz takozvane *crossbar switch*-eve. Godine 1967. Amdahl i Slotnik su objavili svoje diskusije o praktičnosti paralelnog procesiranja na Konferenciji društva "Američke federacije za procesiranje". Zapravo, tokom ove diskusije je Amdahl-ov zakon definisao ograničenje koje paralelizam može da postigne po pitanju ubrzavanja izvođenja programa. Godine 1969. američka kompanija Hanivel (Honeywell) je uvela simetrični multiprocesorski sistem (Multics) sposoban da paralelno izvršava zadatke na 8 procesora, posle čega sledi produkt Kernegi Melon Univerziteta sa C.mmp multiprocesorom. Godine 1964. Slotnik je predložio da se naprave masivno paralelni računari za Lorenc Livermore Nacionalnu Laboratoriju. Izgradnju je finansiralo Američko ratno vazduhoplovstvo i nazvan je ILLIAC IV. Ključna stvar kod dizajna ovog računarskog sistema je bio visok nivo

paralelizacije postignut pomoću 256 procesora koji su omogućavali da mašina radi sa velikim brojem ulaznih podataka. Interesantno je spomenuti da se na kraju za izgradnju ILLIAC IV uložilo 4 puta više para nego što je to bilo predviđeno i da je posle 11 godina tek pušten u rad. Tada je već bio nadmašen od strane jeftinijih komercijalnih superračunara, 1976. godine, kao što je Cray-1.

## A.2 Pajplajning

Pajplajning (eng. *pipelining*) predstavlja način izvođenja instrukcija u procesoru eksploatacijom maksimalnog broja procesorskih komponenti. Proces se najbolje može opisati modelom "pokretne trake" gde se u svakoj fazi izvršava određeni zadatak ali ne pre završetka prethodne faze. Uzmimo kao primer proizvodnju automobila na pokretnoj traci za čiju gradnju je potrebno četiri faze. U prvoj fazi se pravi recimo osnova automobila, pa se u drugoj nadograđuje unutrašnjost i tako dalje. Dok se jedan automobil nalazi u drugoj fazi, drugi automobil započinje svoju izgradnju u prvoj fazi. Tako da se pri proizvodnji drugog automobila ne mora sačekati kraj proizvodnje prvog i na taj način se ne gubi vreme.

## Nivoi paralelizma

Postoji više različitih vrsta paralelnog računanja koje razlikujemo po nivoima na kojima se izvršavanje obavlja, pri čemu pod nivoom podrazumevamo skup operacija:

- Paralelizam na nivou zadataka - svaki od zadataka se izvršava konkurentno i to na istom ili različitom skupu podataka što omogućuje nezavisno izvršavanje zadataka na više procesora.
- Paralelizam na nivou podataka - karakteristično za petlje u kojima su podzadaci nezavisni. U tom slučaju se paralelizacija odvija tako što se isti podzadatak izvršava na različitom skupu podataka.
- Paralelizam na nivou instrukcija - može da se izvede na jednoprocorskim računarima. Instrukcije se regrupišu kako bi se u procesoru izvršavale konkurentno.
- Paralelizam na nivou bitova - izvodi se na jednoprocorskim računarima. Osnovne aritmetičke operacije se regrupišu kako bi se u procesoru izvršavale konkurentno.

## A.3 Prosleđivanje poruka, MPI i PVM

Prosleđivanje poruka (*message passing*) je programska paradigma koja se koristi za komunikaciju između procesa kod određenih klasa paralelnih i distributivnih

okruženja, naročito mašina sa distributivnom memorijom. MPI je postao standard za komunikaciju i može da se realizuje na odvojenim procesorima koji su povezani (brzom ili sporom) mrežom. Zbog ovoga ovaj standard odgovara svakom hardveru današnjih superkompjuteru.

PVM (Parallel Virtual Maschine) je poznato kao distributivno okruženje i sistem za razmenu poruka razvijen 1989. godine. Korišćenjem ovog okruženja se generisala potreba za razvojem standarda za sisteme za razmenu poruka. PVM je dizajniran tako da dozvoljava povezivanje računara koji imaju različite operativne sisteme.

### A.3.1 OpenMP

OpenMP je skup kompajlerskih direktiva. On pomaže kompajleru da generiše višenitne (eng. *multithreaded code*) kodove na višeprosorskim sistemima unutar deljene memorije.

## A.4 MPI Tipovi podataka

Tipovi podataka koje koristimo u MPI funkcijama najčešće odgovaraju osnovnim tipovima podataka programskog jezika Fortran, sa izuzecima za tipove MPI\_BYTE i MPI\_PACKED.

MPI TIP	FORTTRAN
MPI_INTEGER	INTEGER
MPI_REAL	REAL
MPI_DOUBLE_PRECISION	DOUBLE PRECISION
MPI_COMPLEX	COMPLEX
MPI_LOGICAL	LOGICAL
MPI_CHARACTER	CHARACTER(1)
MPI_BYTE	
MPI_PACKED	

Tabela A.1: MPI tip podataka za Fortran

Odgovarajući argumenti za tipove podataka u MPI funkcijama koji se koriste za C programski jezik date su u tabeli A.2.



<b>MPI TIP</b>	<b>C</b>
MPI_SHORT	signed short int
MPI_INT	signed int
MPI_LONG	signed long int
MPI_CHAR	signed char
MPI_UNSIGNED_CHAR	unsigned char
MPI_UNSIGNED_SHORT	unsigned short int
MPI_UNSIGNED	unsigned int
MPI_UNSIGNED_LONG	unsigned long int
MPI_FLOAT	float
MPI_DOUBLE	double
MPI_LONG_DOUBLE	long double
MPI_BYTE	
MPI_PACKED	

Tabela A.2: MPI tip podataka za C

## A.5 Klaster

Klasterski sistem je skup računara koji tesno sarađuju u izvršavanju određenih zadataka tako što funkcionišu kao jedan računar. Čvorovi klastera su često povezani brzom LAN mrežom u cilju povećanja performansi i/ili dostupnosti u poređenju sa jednim računarom koji izvršava iste zadatke.

Klaster mora da zadovolji sledeće kriterijume:

1. Nadogradivost;
2. Dostupnost;
3. Celovitost sistema;
4. Brzu komunikaciju;
5. Programabilnost;
6. Primenljivost.

### Nadogradivost

Klaster sistemi moraju da imaju mogućnost dinamičke i transparentne nadogradnje. To znači da dodavanje ili uklanjanje čvorova iz klastera ne sme biti vidljivo korisniku i uticati na izvođenje aplikacija.

## Dostupnost

Ukoliko čvor klastera postane privremeno ili trajno nedostupan njegove zadatke preuzimaju ostali dostupni čvorovi.

## Celovitost sistema

Celovitost sistema je slika kreirana na onom nivou na kome se grupa resursa predstavlja kao jedan snažniji resurs. Svaki klaster sistem koji ima ovu osobinu deluje kao jedan računar.

## Brza komunikacija

Čvorovi klastera su povezani brzom računskom mrežom i koriste se protokoli koji omogućavaju brzu komunikaciju.

## Programabilnost

Klasteri moraju da podrže razvoj prenosivih aplikacija. Ovo se omogućuje pomoću API (eng. Application Programming Interface) alata koji sakrivaju arhitekturu klastera.

## Primenljivost

Tokom izgradnje klastera potrebno je izbegavati dizajn koji bi mogao da ograniči njegovu primenu i osigurati primenljivost na širokom spektru problema.

## A.6 Transpjuteri

### Karakteristike paralelnog programiranja na transpjuterima

Vreme potrebno za prenos jednog podatka između dva procesora je oko četiri puta duže u odnosu na aritmetičku operaciju na jednom procesoru, nad tim istim podatkom. Intuitivno se nameće ideja da treba smanjiti komunikaciju. Naravno, što više smanjujemo komunikaciju, smanjuje se i razmena podataka, pa se smanjuje i iskorišćenost procesora. Prema tome treba naći kompromis između smanjenja komunikacije i smanjenja iskorišćenosti procesora.

Problemi pogodni za paralelizaciju na transpjuterima su, uglavnom, oni sa velikim brojem računarskih operacija, a malim brojem ulazno-izlaznih operacija. Problemi sa velikim brojem ulazno-izlaznih operacija, a malim brojem računarskih operacija, nisu pogodni za paralelizaciju na transpjuterima. Razlog je mogućnost paralelizacije računskih operacija, uz nemogućnost paralelizacije ulazno-izlaznih operacija. Na primer, ulaz i izlaz podataka je direktno dopušten samo na prvom transpjuteru,

a ostalim transpjuterima su dozvoljene samo indirektno poruke od prvog. Prvi transpjuter, kao posrednik, tada izvrši odgovarajuće ulazno-izlazne operacije.

S obzirom na vreme trajanja komunikacije, u slučaju velikog broja ulazno-izlaznih podataka, ulaz ili izlaz podataka može trajati mnogo duže od samog izračunavanja. Postoje neki problemi gde same ulazno-izlazne operacije i odgovarajuće komunikacije dugo traju, pa u praksi izvršavanje na više procesora ne ubrzava izvršavanje programa. Ponekad, zbog tehničkih ograničenja ili brzine komunikacije, algoritmi koji su teorijski jako efikasni, u praksi pokazuju loše rezultate.

Imajući u vidu data ograničenja, na transpjuterima se mogu uspešno paralelizovati, uglavnom, oni problemi gde nema mnogo ulaznih i izlaznih podataka, ali ima puno računanja na njima. U većini takvih slučajeva se problem može uspešno podeliti na potprobleme, koji se mogu izvršavati nezavisno, odnosno paralelno. Naravno, to je samo neophodan, ali ne i dovoljan uslov da bi se problem mogao uspešno paralelizovati na transpjuterima.

Tehnička ograničenja transpjutera su ([20]):

1. Svaki transpjuter može biti povezan sa najviše četiri druga transpjutera.
2. Svaki transpjuter mora biti resetovan od nekog transpjutera, preko jedne od četiri veze. Jedino prvi transpjuter je resetovan od računara domaćina.
3. Bilo koji transpjuter može resetovati najviše dva transpjutera.

Transpjutere možemo podeliti u tri grupe ([20]):

1. Prvi transpjuter.
2. Transpjuteri, osim prvog, koji imaju "naslednike".
3. Transpjuteri koji nemaju "naslednike".

### A.6.1 Masivni višejezgarni sistemi

Vvišejezgarni sistemi danas omogućavaju da se programi izvršavaju 10 do 100 puta brže nego na klasičnim procesorima (sa jednom centralnom jedinicom, CPU). Razvijanje aplikacija koje će maksimalno koristiti višejezgarni sistem je teško i zahteva od programera da za pisanje sekvencijalnih programa pređu na pisanje paralelnih programa. Aplikacije bi trebalo da se prilagođavaju različitim strukturama sistema, u zavisnosti od broja jezgara.

Spajanje višejezgarinog hardvera sa paralelnim računanjem podstaknuto operativnim sistemom, platformom i alatima omogućuje: prirodne jezike, sofisticirane analize slika i manipulacije nad njima, semantičku analizu podataka i drugo.

## B.1 Optička dužina puta

Razlika između pređenih puteva izražena preko talasne dužine svetlosti ne zavisi samo od geometrijske dužine puta već i od indeksa prelamanja sredine kroz koju se talas prostire. Da bi opisali ovu razliku uvodi se *optička dužina puta*. Optička dužina puta talasa koji je prešao geometrijski put dužine  $d$  kroz sredinu sa konstantnim indeksom prelamanja  $n$  je jednako  $nd$ .

Ako se talas prostire kroz sredine sa različitim indeksima prelamanja tada optičku dužinu puta računamo preko integrala

$$\int_C n(s) ds, \quad (\text{B.1})$$

gde je  $n(s)$  ukupan indeks prelamanja dat kao funkcija od rastojanja  $s$  duž trajektorije opisane krivom  $C$ .

## B.2 Kritične krive i kaustici

### B.2.1 Dokaz za kritične krive

Neka je  $\vec{\theta}(\lambda)$  parametrizacija kritične krive; tada je kaustik definisan sa  $\vec{\beta}(\vec{\theta}(\lambda))$ . Vektor tangente kritične krive je izvod  $\dot{\vec{\theta}}(\lambda) \equiv \frac{d\vec{\theta}(\lambda)}{d\lambda}$ , a odgovarajući vektor tangente kaustika je

$$\frac{d\vec{\beta}(\vec{\theta}(\lambda))}{d\lambda} = \frac{\partial \vec{\beta}}{\partial \vec{\theta}} \frac{d\vec{\theta}}{d\lambda} = A(\vec{\theta}(\lambda)) \dot{\vec{\theta}}(\lambda).$$

Ukoliko je  $\vec{\theta}(\lambda)$  paralelan sopstvenom vektoru matrice  $A$  u slučaju  $\det A = 0$ , tada je odgovarajući vektor tangente kaustika nula-vektor. Drugim rečima, ako je pravac vektora tangente kritične krive određen singularnim pravcem od  $A$ , tada kaustik ne mora biti glatka kriva. U tom slučaju kaustika ima vrh ili šiljak (eng. *cusp*). Delovi kaustika koji nisu šiljci su glatke krive i nazivamo ih *kaustik sa prevojem* (eng. *fold caustic*).

## B.2.2 Broj likova

Za sočiva koja nemaju singularitete (tj. tačkaste mase) broj likova je jedan ukoliko je (ugaoni) položaj izvora dovoljno daleko od sočiva (ili sistema sočiva), te se broj likova koji odgovaraju regionima u ravni izvora, lako može odrediti kada znamo gde su kaustici.

Ukoliko je izdužen izvor smešten na kaustiku tada možemo da istaknemo dva specifična položaja:

- kada se izvor nalazi na prevoju generišu se dva lika
- kada se izvor nalazi na šiljku generišu se tri lika.

U svakom slučaju, samo će se za deo izvora u unutrašnjem regionu kaustika formirati višestruki likovi.

## B.2.3 Vremensko zakašnjenje (eng. *time delay*)

Savijanje zraka svetlosti pod uticajem gravitacionog polja mase pored koje zrak prolazi dovodi do kašnjenja u vremenu za koje je svetlosti potrebno da dopre od izvora do posmatrača. Ovo vremensko kašnjenje ima dve komponente:

$$t = t_{geom} + t_{grav}. \quad (\text{B.2})$$

Put zraka koji se savija je duži i time je svetlosti potrebno više vremena da taj put pređe nego putujući duž pravog zraka. S obzirom da se svaki svetlosni zrak savija pod drugim uglom, njihove geometrijske dužine su drugačije što dovodi do "geometrijskog vremenskog zakašnjenja"  $t_{geom}$  između savijenog i pravog zraka. Druga komponenta  $t_{grav}$  je vremensko kašnjenje koje nastaje usled uticaja gravitacije, zbog čega dolazi do "usporenja" fotona koji putuju kroz gravitaciono polje sočiva i ovaj efekat je poznat pod nazivom *Šapiro efekat*<sup>1</sup>. S obzirom da se gravitaciono vreme 'dešava' na crvenom pomaku sočiva  $z_l$ , ono u sebi sadrži dodatni faktor  $(1 + z_l)$  koji opisuje dilataciju vremena.

---

<sup>1</sup> Eng. Shapiro effect