VILNIAUS UNIVERSITETAS
MATEMATIKOS IR INFORMATIKOS FAKULTETAS
INFORMATIKOS KATEDRA

LYGIAGRETIEJI ALGORITMAI

Collatz'o ilgiausios iteracijos paieškos algoritmo analizė

Atliko: 3 kurso, 2 grupės studentas
Tomaš Maconko

Vilnius
2012

Užduotis: Ilgiausios Collatz'o iteracijos paieška duotajame skaičių intervale.
Duota: Skaičių rėžiai nuo M iki N.
Rezultatas: Iteracijos ilgis.

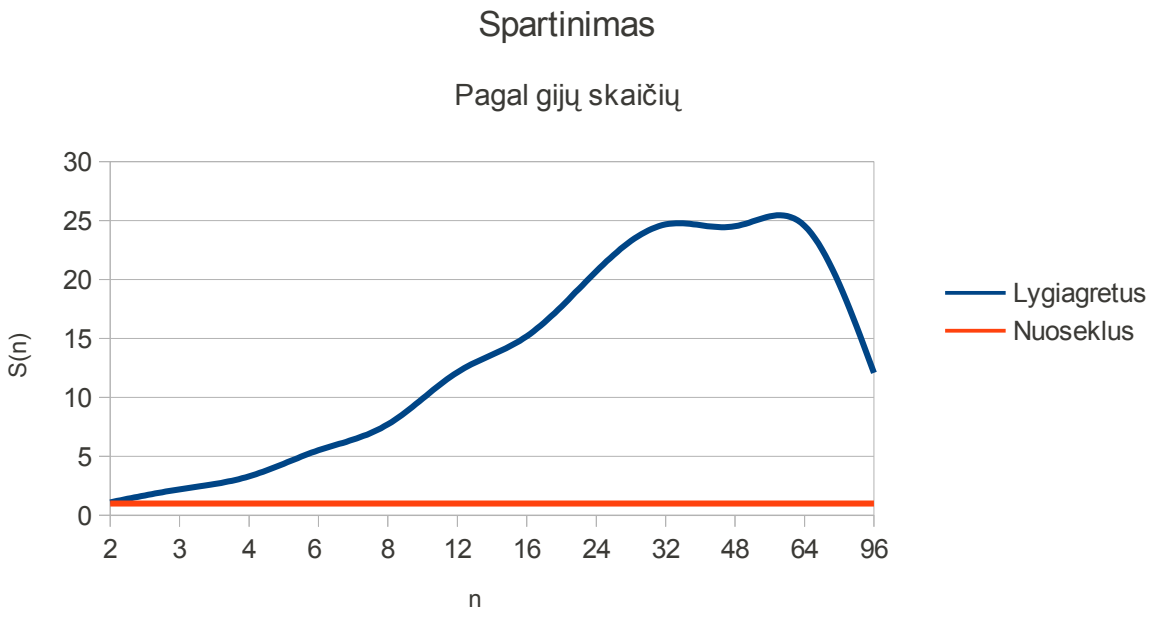
Sprendimas: Užduotis buvo spręsta MPI lygiagretaus programavimo interfeisu. Buvo išlygiagretintas paprastas algoritmas, t. y. paprastas ciklas „for“ buvo perrašytas į MPI tarpprocesorinį bendravimą – „MASTER-SLAVE“ struktūrą.
Programos paleidimo metu reikia įvesti 5 argumentus: *[M] [N] [Chunk] [OutputFlag]*, kur:

- [M] ir [N] – skaičių intervalo rėžiai;
- [ChunkSize] – dinaminio darbo dalijimosi bloko dydis;
- [OutputFlag] – 0: išvesti į ekraną tik rezultatą, 1: išvesti į ekraną kiekvieną iteraciją.

Testavimas: Visi testai buvo atlikti Vilniaus Universiteto „Paskirstytų skaičiavimų tinkle“, Debian GNU/Linux operacinės sistemos (OS) versijos 6 terpėje, „short“ particijoje.

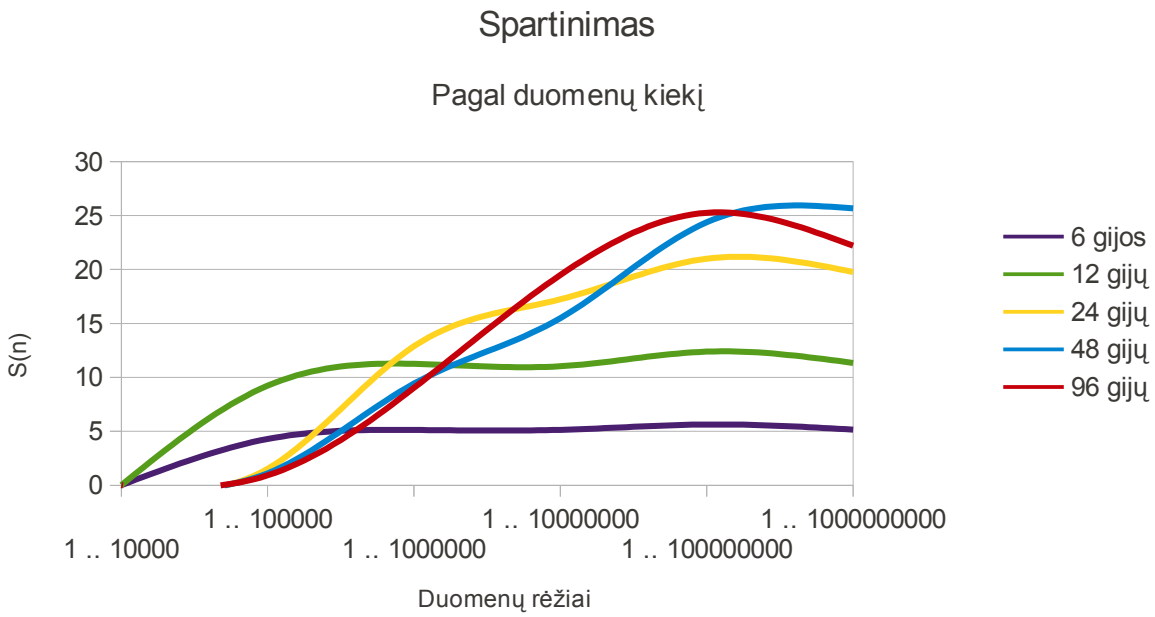
Testas 1: Spartinimo priklausomybė nuo gijų skaičiaus.

Šis testas parodo spartinimo didėjimą nuo didėjančio gijų skaičiaus. Spartinimo kitimo grafiką matome pav. 1. Kadangi ši MPI realizacija reikalauja minimum dviejų branduolių, tai negalime nieko nuspėti, kas būtų jei būtų tik viena gija. Pradedama nuo 2 ir čia matoma, jog spartinimas yra arti 1. Toliau spartinimas pradeda augti maždaug tiesine funkcija, kol nepasiekia maždaug 32 gijų skaičiaus. Tada spartinimas baigia augti ir didėjant gijų skaičiui nuo 32 iki 64 beveik nesikeičia. Vėliau pradeda sparčiai mažėti. Iš to galima sekti, jog optimaliausiai galima išnaudoti 32 gijas tokiam uždaviniui su panašiu duomenų kiekiu. Šis testas yra vykdomas kai <chunkSize> yra lygus 100, o duomenų rėžiai yra nuo 1 iki 100000000.



Pav. 1: Spartinimo priklausomybė nuo gijų skaičiaus.

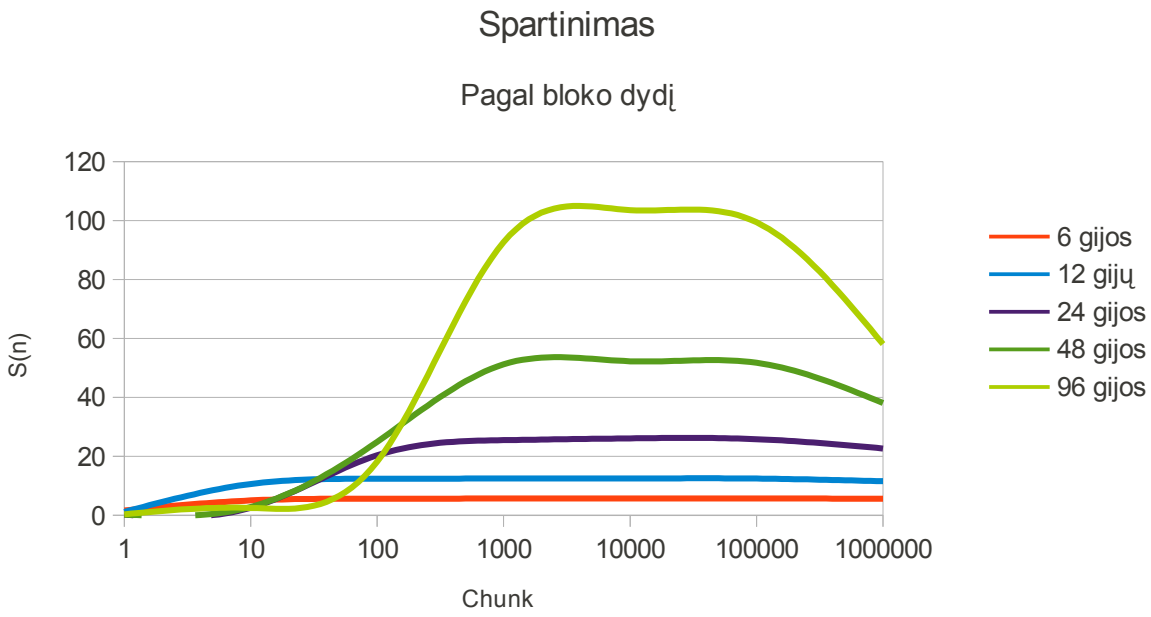
Testas 2: Spartinimo priklausomybė nuo duomenų kiekio (plečiamumas).
Šis testas parodo algoritmo plečiamumą, t. y. spartinimo priklausomybė nuo duomenų kiekio. Šiuo atveju - tai yra M ir N skaičiai nurodantys rėžius. Plečiamumo grafiką galima pamatyti pav. 2. Grafike yra pavaizduoti duomenys iš 5 skirtingų atvejų, kai duomenys keičiasi esant 6 gijoms, 12, 24, 48 ir 96. Iš grafiko matome, jog didėjant duomenų kiekiui dydžiui spartinimas svyruojant auga. Augimo viršunė yra pasiekiama, kai M yra lygus 1, o N yra arti 100000000. Šioje vietoje beveik visos gijos įgauna super-linearų spartinimą. Šis testas buvo atliekamas, kai <chunkSize> buvo konstanta lygi 100.



Pav. 2: Algoritmo plečiamumas.

Testas 3: Spartinimo priklausomybė nuo bloko dydžio.

Šis testas parodo algoritmo spartinimą, kuris priklauso nuo besikeičiančio bloko („chunk“) dydžio. Šis kintamasis nurodo lygiagretaus darbo pasidalijimą tarp gijų, kuris yra dinaminis. Spartinimo kitimo grafiką galima pamatyti pav. 3. Dideliam gijų skaičiui nėra reikalo parinkti mažą bloko dydį, kadangi gaunasi daug pranešimų tarp procesų, kurie sulėtina darbą. Dideli bloko dydžiai irgi nėra pageidaujami, kadangi atsiranda procesų prastovos, dėl ko spartinimas mažėja. Iš grafiko matome, jog optimalus bloko dydis yra pradedant nuo 1000 ir baigiant 100000. Šioje vietoje, kaip galima pastebėti, atsiranda super-linearus spartinimas. Šis testas buvo atliktas esant duomenų rėžiams nuo 1 iki 100000000.



Pav. 3: Spartinimo priklausomybė nuo bloko dydžio.

Išvados

Iš aukščiau aprašytų testų galima nuspręsti, jog Collatz'o problemos realizacijos MPI pagalba spartinimas priklauso nuo:

- Gijų skaičiaus (yra ribiniai/optimalūs gijų skaičiai);
- Duomenų kiekio ir dydžio (didesniam, bet ne per dideliam, duomenų kiekiui spartinimas auga);
- Bloko dydžio (nuo mažų bloko dydžių spartinimas yra mažas, vėliau didėjant iki ribinio taško spartinimas auga ir po to mažėja);

Iš aukščiau aprašytų testų aprašų, galima nuspręsti, jog ši algoritmo realizacija yra sekminga, kadangi pagal daugumą kreivės taškų pastebima, jog algoritmas įgija ideliai-linearų spartinimą, o kai kuriuose vietose netgi – super-linearų spartinimą.