**O'ZBEKISTON RESPUBLIKASI AXBOROT TEXNOLOGIYALARI VA KOMMUNIKATSIYALARINI RIVOJLANTIRISH VAZIRLIGI MUHAMMAD AL-XORAZMIY NOMIDAGI TOSHKENT AXBOROT TEXNOLOGIYALARI UNIVERSITETI 2022**

**Abdullayev Abbosjon Xasanboy o’g’li**

**Sonli usullar uchun parallel algoritmlar va OpenMp kutubxonasidan foydalanib parallel hisoblash dasturiy ta’minotini yaratish.**

**5A330501– Kompyuter injinirigi: (Axborot va multimedia texnologiyalari) mutaxassisligi bo‘yicha**

**‎‎**

**Magistr akadеmik darajasini olish uchun yozilgan**

**DISSЕRTATSIYA**

**Ilmiy rahbar: f-m.f.n, dotsent Ne’matov A.**

**Toshkent 2022**

**Annotatsiya**

Ushbu dissertatsiya ishida sonli usullar uchun parallel algoritmlar ishlab chiqish, parallel algoritmalarning afzalliklari va kamchiliklari, ularning ketma-ket algoritmlardan farqlari tushuntirilgan. Hamda Parallel algoritmlarning asosiy ishlash ko'rsatkichlari ketma-ket algoritmlar bilan taqqoslandi va OpenMP kutubxonasidan foydalanish tartibi ko‘rsatib o‘tilgan.

**Annotation**

In this dissertation work the development of parallel algorithms for numerical methods, the advantages and disadvantages of parallel algorithms, their differences from serial algorithms are explained. Also, the basic performance of Parallel Algorithms is compared with serial algorithms, and the order of use of the OpenMP library is shown.

**Mundarija**

[Kirish 4](#_Toc106345305)

[I BOB Parallel algoritmlar va parallel hisobllash. 8](#_Toc106345306)

[1.1 Parallel hisoblash qanday ishlaydi. 8](#_Toc106345307)

[1.2 Parallel dasturlash nima uchun ishlatiladi? 10](#_Toc106345308)

[1.3 Parallel arxitekturalar 12](#_Toc106345309)

[1.4 Parallel hisoblashning afzallilklari va kamchiliklari. 15](#_Toc106345310)

[1.5 Adabiyotlar tahlili va metodologiya. 16](#_Toc106345311)

# Kirish

Arxitekturani takomillashtirish bilan bir qatorda parallel kompyuterlarning dasturiy ta'minoti ham rivojlanib bormoqda. Amaliyot shuni ko'rsatdiki, parallel hisoblash tizimlari apparat va dasturiy ta'minot qismlarining rivojlanishini birbiridan alohida-alohida ko'rib bo'lmaydi. Qismlarning biridagi yangilik, ikkinchisidagi o'zgarishga olib keladi. Hozirda bizni avvalo parallel dasturlash texnologiyalari sohasidagi o'zgarishlar qiziqtiradi. Albatta, zamonaviy dasturiy ta'minot ishlab chiqish ixtiyorida nafaqat Fortran yoki assembler bor, balki ko'p boshqa tizimlar va dasturlash tillari yaratildi. Shu bilan birga, ayni paytda, samarali parallel dasturiy ta'minot ishlab chiqish muammosi umumiy parallel hisoblashning asosiy muammosiga aylandi.

Ta’kidlash joizki, maxsus izohlardan foydalanish nafaqat parallel [bajarish imkoniyatini beradi](http://hozir.org/9-amaliy-mashgulot-mathcad-dasturida-olchamli-grafiklar-qurish.html), balki dasturning asl versiyasini to’liq saqlab qoladi. Amalda, u juda qulay - kompilyator parallelizm haqida hech narsa bilmasa, barcha mahsus izohlarni semantik ketma ketlikni saqlab qolgan holda o’tkazib yuboradi.  
Parallel dasturlar tuzish uchun izohlardan foydalanishdan tashqari, tez-tez mavjud dasturlash tillari kengaytirishni qo’llashadi. Qo’shimcha operatorlar va foydalanuvchiga dasturning parallel tuzilmasini aniq belgilash va ayrim hollarda parallel dasturni bajarishni boshqarish imkonini beruvchi o’zgaruvchilarni ifodalovchi yangi elementlar kiritiladi. Shunday qilib, High Performance Fortran (HPF) til, an'anaviy FORTRAN operatorlari va mahsus izohlar tizimiga qo'shimcha ravishda, dasturning parallel sikllarini ifodalash uchun kiritilgan yangi forall operatorini o`z ichiga oladi.

Agar parallel tizimlar arxitekturasi o’ziga hosliligini, yoki ayrim mavzu doirasining ba’zi bir vazifa turining hususiyatlarini aniq ifodalash kerak bo’lsa, unda parallel dasturlashning mahsus tillaridan foydalaniladi.. Transputer tizimlarini dasturlash uchun [Occam tili tashkil etilgan](http://hozir.org/inson-va-tuproq-kitobi-jayron-ekologik-markazi.html), Konveyer mashinalarini dasturlash uchun, birlamchi belgilash tili Sisal loyihalashtirilgan.

Parallel hisoblash - bu bir vaqtning o'zida bir nechta operatsiyalarni bajarish. Parallel hisoblashdan to'liq foydalanish avtomatik ravishda sodir bo'lmaydi. Bu dasturchidan biroz kuch talab qiladi. Birinchidan, siz ilovada parallellik potentsialini aniqlashingiz va ochib berishingiz kerak. Potentsial parallellik yoki parallellik, tizimda resurslar mavjud bo'lganda, operatsiyalarni istalgan tartibda bajarish xavfsiz ekanligini tasdiqlaganingizni bildiradi. Parallel hisoblash uchun qo'shimcha talab mavjud; bu operatsiyalar bir vaqtning o'zida amalga oshirilishi kerak. Buning amalga oshishi uchun siz ularni bir vaqtning o'zida bajarish uchun resurslardan to'g'ri foydalanishingiz kerak. Parallel hisoblash seriyali dunyoda mavjud bo'lmagan yangi tashvishlarni keltirib chiqaradi. Parallel bajarishning qo'shimcha murakkabliklariga moslashish uchun fikrlash jarayonidagi o'zgarish kerak, ammo amaliyot bilan u ikkinchi tabiatga aylana boshlaydi. . Parallel hisoblash tizimida bir yoki bir necha hisoblash tugunlarida bir yoki bir nechta dasturiy komponentlari bo'lishi mumkin. Tizim bir kompyuter tizimi sifatida paydo bulishi uchun, bu qismlar o'zaro aloqalarni qo'llab-quvvatlashi kerak. Oraliq dasturiy ta’minotning asosiy roli shundaki, ushbu vazifalarni soddalashtirish va parallel taqsimlangan hisoblash-tizimlari komponentlarini o'zaro ta’sirini qurilishida amaliy dasturchilar tomonidan qo’llaniladigan abstrakstiyalarni ta’minlashdir.

Hozirgi zamonda texnikalarni, dasturlarni mavqeyi, asosan uning ishlash tezligiga qarab belgilanadi. Agar ularning tezligi yuqori bo’lsa, unga bo’lgan talab ham yuqori bo’ladi. Aksincha, jarayonlarni amalga oshirish uchun ketadigan vaqt qanchalik sekin davom etadigan bo’lsa, unga bo’lgan qiziquvchilarning soni ham shuncha past bo’ladi va natijada dastur yoki texnikaning sifati anchagina pastlaydi. Tezlik hozirgi zamonning eng muhim talablaridan hisoblanadi.

**Dissertasiya mavzusining asoslanishi va uning dolzarbligi:** Bu ishda ikki vazifa – algoritmlarning dastur effektivligiga qanday ta’sir etishi va turli xil algoritmlarning analizini o’rganishdir. Ba’zi bir zamonaviy dasturiy ta’minotlarga e’tibor qilsak, ularning ayrim tuzuvchilari na dasturning ishlash effektivligiga va na xotiraning aql bilan ishlatilishiga e’tibor qilishadi. Ularning fikricha, dastur ko’p joy olsa, foydalanuvchi qo’shimcha xotira sotib olishga majbur bo’ladi yoki yangi tezroq ishlaydigan komyuter sotib oladi.

Parallel ishlov berish - bu umumiy vazifaning alohida qismlarini bajarish uchun ishlaydigan ikki yoki undan ortiq [protsessorlarni](https://www.techtarget.com/whatis/definition/processor) (CPU) hisoblash usuli. Bir nechta protsessorlar orasida vazifaning turli qismlarini ajratish dasturni ishga tushirish vaqtini qisqartirishga yordam beradi. Bir nechta protsessorga ega har qanday tizim parallel ishlov berishga, shuningdek, bugungi kunda kompyuterlarda keng tarqalgan [ko'p yadroli protsessorlardan foydalanish aimkon berishi mumkin.](https://www.techtarget.com/searchdatacenter/definition/multi-core-processor)

Ko'p yadroli protsessorlar - bu yaxshi ishlashi, quvvat sarfini kamaytirish va bir nechta vazifalarni yanada samarali qayta ishlash uchun ikki yoki undan ortiq protsessorlarni o'z ichiga olgan IC chiplari. Ushbu ko'p yadroli sozlashlar bir xil kompyuterda o'rnatilgan bir nechta alohida protsessorlarga o'xshaydi. Ko'pgina kompyuterlar ikkitadan to'rttagacha yadroga ega bo'lishi mumkin; 12 yadrogacha ko'tariladi.

Parallel ishlov berish odatda murakkab vazifalar va hisob-kitoblarni bajarish uchun ishlatiladi. Ma'lumotlar olimlari odatda hisoblash va ma'lumotlarni ko'p talab qiladigan vazifalar uchun parallel ishlov berishdan foydalanadilar.

**Tadqiqotning maqsadi:** Sonli usullar uchun parallel algoritmlar ishlab chiqish va ushbu algoritmlar asosida dasturiy ta’minot yaratish.

**Dissertatsiya ishining vazifalari:**

* Sonli usullar uchun parallel algoritmlarni tahlil qilish.
* Yangi parallel algoritmlar yaratish.
* Parallel algoritmlarni OpenMP kutubxonasidan foydalanib ishlatish.
* Parallel algoritmlarni dasturiy ta’minotda to‘g‘ri qo‘llay olish.
* Parallel hisoblash dasturiy ta’minotini yaratish.

**Dissertatsiya ishining obyekti:** Parallel hisoblashlarni amalga oshiruvchi OpenMP kutubxonasi.

**Dissertatsiya ishining predmeti:** Sonli usullar va hisoblash uchun parallel algoritmlar.

**Tadqiqotning ilmiy- uslubiy yangiligi:** Parallel hisoblash - bu bitta masalani hal qilish uchun ikki yoki undan ortiq protsessorlardan (yadro, kompyuterlar) birgalikda foydalanish. Bu hisoblash arxitekturasining bir turi bo'lib, unda bir nechta protsessorlar bir vaqtning o'zida dastur yoki hisoblashni bajaradi yoki qayta ishlaydi. Parallel hisoblash bir vaqtning o'zida bir nechta protsessorlar o'rtasida ish yukini taqsimlash orqali katta hisob-kitoblarni amalga oshirishga yordam beradi, ularning barchasi bir vaqtning o'zida hisoblash orqali ishlaydi.

U odatda katta hisoblash yoki ishlov berish quvvatini talab qiladigan operatsion muhitda/stsenariylarda qo'llaniladi.

Tadqiqot mavzusi bo’yicha adabiyotlar sharhi (tahlili): 1958 yil aprel oyida S. Gill (Ferranti) parallel dasturlash va tarmoqlanish va kutish zarurligini muhokama qildi. Shuningdek, 1958 yilda IBM tadqiqotchilari Jon Kok va Daniel Slotnik birinchi marta raqamli hisob-kitoblarda parallellikdan foydalanishni muhokama qilishdi. Burroughs korporatsiyasi 1962 yilda D825 ni taqdim etdi, bu to'rt protsessorli kompyuter bo'lib, 16 tagacha xotira moduliga o'zaro faoliyat kalit orqali kirgan.

Tarixiy parallel hisoblashlar ilmiy hisoblash va ilmiy muammolarni simulyatsiya qilish uchun, xususan, meteorologiya kabi tabiiy va muhandislik fanlarida ishlatilgan. Bu parallel apparat va dasturiy ta'minotni, shuningdek, yuqori unumli hisoblashni loyihalashga olib keldi.

**Tadqiqotning asosiy masalalari va farazlari:** Asosiy markaziy protsessor (CPU yoki protsessor) quvvat sarfi va haddan tashqari qizib ketish muammosini hal qilish uchun ishlab chiqaruvchilar bir nechta yadroli energiya tejaydigan protsessorlarni ishlab chiqarishni boshladilar. Yadro protsessorning hisoblash birligi bo'lib, ko'p yadroli protsessorlarda har bir yadro mustaqildir va bir vaqtning o'zida bir xil xotiraga kira oladi. Ko'p yadroli protsessorlar ish stoli kompyuterlariga parallel hisoblashni olib keldi. Shunday qilib, ketma-ket dasturlarni parallellashtirish asosiy dasturlash vazifasiga aylandi.

# I BOB Parallel algoritmlar va parallel hisobllash.

## 1.1 Parallel hisoblash qanday ishlaydi.

Parallel dasturlash turli tugunlar yoki yadrolarga vazifalarni belgilash orqali ishlaydi. High Performance Computing (HPC) tizimlarida tugun operatsion tizimda ishlaydigan xotira va protsessorlarni o'z ichiga olgan kompyuter tizimining mustaqil birligidir. Markaziy protsessorlar (CPU) va [grafik ishlov berish birliklari (GPU)](https://totalview.io/blog/gpu-computing) kabi protsessorlar yadrolar to'plamini o'z ichiga olgan chiplardir. Yadrolar - buyruqlarni bajaruvchi birliklar; protsessorda bir nechta yadro va tugunda bir nechta protsessor bo'lishi mumkin.

Odatda kompyuter olimi murakkab vazifani dasturiy vosita yordamida bir nechta qismlarga ajratadi va har bir qismni protsessorga tayinlaydi, keyin har bir protsessor o’z qismini hal qiladi va ma’lumotlar yechimni o’qish yoki vazifani bajarish uchun dasturiy vosita tomonidan qayta yig’iladi.

Odatda har bir protsessor normal ishlaydi va ko’rsatmalarga muvofiq parallel ravishda operatsiyalarni bajaradi va kompyuter xotirasidan ma’lumotlarni oladi. Protsessorlar, shuningdek, bir-biri bilan aloqa qilish uchun dasturiy ta’minotga tayanadi, shuning uchun ular ma’lumotlar qiymatlaridagi o’zgarishlar bo’yicha sinxronlasha oladilar. Agar barcha protsessorlar bir-biri bilan sinxronlashtirilsa, vazifa oxirida dasturiy ta’minot barcha ma’lumotlar qismlarini bir-biriga moslashtiradi.

Parallel dasturlash yordamida dasturchi o’z dasturlarini bir nechta tugunlar yoki protsessorlarda ishlatishni osonlashtirish uchun maxsus dasturiy ta’minot bilan kod yozadi. Qayta ishlashni tezlashtirish uchun parallel dasturlashdan foydalanish mumkinligiga oddiy misol tasvirni qayta ranglashdir. Ishlab chiquvchi tasvirni teng qismlarga bo’lish orqali tasvirning individual tomonlarini o’zgartirish bo’yicha umumiy vazifani ajratish uchun kod yozadi va keyin har bir qismning rangini o’zgartirishni har biri o’z hisoblash resurslarida ishlaydigan boshqa parallel vazifaga tayinlaydi. Parallel vazifalar bajarilgandan so’ng, to’liq tasvir qayta yig’iladi.

Parallel ishlov berish usullari o’rnatilgan, mobil, noutbuklar va ish stantsiyalaridan tortib dunyodagi eng yirik superkompyuterlargacha bo’lgan qurilmalarda qo’llanilishi mumkin. Turli xil kompyuter tillari parallellikni ta’minlash uchun turli xil texnologiyalarni taqdim etadi. C, C++ va Fortran uchun OpenMP, ochiq multi-processing, protsessor yadrolari bo’ylab parallel vazifalarni bajarishga imkon beruvchi parallel ilovalarni ishlab chiqish uchun platformalararo API taqdim etadi. Jarayonlar turli xil kompyuterlar yoki tugunlar o’rtasida bog’lanishi kerak bo’lganda, odatda MPI, xabarlarni uzatish interfeysi kabi texnologiya qo’llaniladi. Ikkala modelning ham afzalliklari bor. Bitta tugundagi bir nechta yadro xotirani ulashadi. Umumiy xotira odatda tarmoq orqali tugunlar o’rtasida o’tadigan xabarlarga qaraganda ma’lumot almashish uchun tezroq. Biroq, bitta tugun qancha yadroga ega bo’lishi mumkinligi chegarasi mavjud. Loyihalar kattalashgani sayin, Ishlab chiquvchilar parallelizmning ikkala turini birgalikda ishlatishlari mumkin. Ishlab chiquvchilar duch keladigan muammolardan biri bu algoritmni to’g’ri parchalash va maksimal ishlash uchun bir nechta tugunlar va bir nechta yadrolar bo’ylab parallellashtirish va to’g’ri ishlamasa, parallel qo’llanilishini tuzatishdir.

Bir nechta protsessorga ega bo’lmagan kompyuterlar, agar ular bir [klaster](https://www.techtarget.com/whatis/definition/cluster) hosil qilish uchun tarmoqqa ulangan bo’lsa, hali ham parallel ishlov berishda ishlatilishi mumkin.

Eng qadimgi kompyuterlarda bir vaqtning o’zida faqat bitta dastur ishlagan. Ishlash uchun bir soat kerak bo’lgan hisoblash intensiv dastur va bir soat davom etadigan lenta nusxalash dasturi jami ikki soat davom etadi. Parallel ishlov berishning dastlabki shakli ikkala dasturni birgalikda bajarishga imkon berdi. Kompyuter kiritish-chiqarish operatsiyasini boshlaydi va operatsiya tugashini kutayotganda, u protsessor talab qiladigan dasturni bajaradi. Ikki ish uchun umumiy bajarilish vaqti bir soatdan sal ko’proqni tashkil qiladi.

Keyingi yaxshilanish ko’p dasturlash edi. Ko’p dasturlash tizimida foydalanuvchilar tomonidan taqdim etilgan bir nechta dasturlarning har biriga protsessordan qisqa vaqt foydalanishga ruxsat berildi. Foydalanuvchilar uchun barcha dasturlar bir vaqtning o’zida bajarilayotgandek tuyuldi. Resurs nizosi muammolari birinchi navbatda ushbu tizimlarda paydo bo’ldi. Resurslarga bo’lgan aniq so’rovlar o’lik muammosiga olib keldi  [,](https://www.techtarget.com/whatis/definition/deadlock) bunda resurslarga bir vaqtning o’zida so’rovlar dasturning manbaga kirishiga to’sqinlik qiladi. Bog’lanish bo’yicha ko’rsatmalarga ega bo’lmagan mashinalarda resurslar uchun raqobat muhim bo’lim tartibiga olib keladi.

Vektorli ishlov berish bir vaqtning o’zida bir nechta ishlarni bajarish orqali samaradorlikni oshirishga qaratilgan yana bir urinish edi. Bunday holda, bitta ko’rsatma yordamida ikkita son massivini qo’shish (yoki ayirish, ko’paytirish yoki boshqa yo’l bilan o’zgartirish) imkonini beradigan qobiliyatlar mashinalarga qo’shildi. Bu ma’lumotlar vektorlar yoki matritsalar shaklida tabiiy ravishda paydo bo’ladigan ba’zi muhandislik ilovalarida qimmatli edi. Kamroq shakllangan ma’lumotlarga ega bo’lgan ilovalarda vektorni qayta ishlash unchalik qimmatli emas edi.

Parallel qayta ishlashning keyingi bosqichi ko’p ishlov berishni joriy qilish  [edi](https://www.techtarget.com/searchdatacenter/definition/multiprocessing) . Ushbu tizimlarda ikki yoki undan ortiq protsessorlar bajarilishi kerak bo’lgan ishni bo’lishdi. Eng qadimgi versiyalarda master/slave konfiguratsiyasi mavjud edi. Bitta protsessor (magistr) tizimdagi barcha ishlar uchun javobgar bo’lishi uchun dasturlashtirilgan; ikkinchisi (qul) faqat xo’jayin tomonidan topshirilgan vazifalarni bajargan. Ushbu tartibga solish zarur edi, chunki keyinchalik mashinalarni tizim resurslarini boshqarishda hamkorlik qilishlari uchun qanday dasturlash kerakligi tushunilmagan edi.

## 1.2 Parallel dasturlash nima uchun ishlatiladi?

Parallel dasturlashning vazifalarni ajratish qobiliyati uni katta hajmdagi ma’lumotlar, murakkab hisoblar yoki katta simulyatsiyalar bilan bog’liq murakkab muammolar uchun mos echimga aylantiradi. Ilgari hal qilib bo’lmaydigan muammolar ob-havo simulyatsiyasi, vaktsinani ishlab chiqish va astrofizika tadqiqotlari kabi parallel dasturlash yordamida parchalangan.

Parallel dasturlashdan foydalanish holatlari quyidagilarni o’z ichiga oladi:

* Amaliy fizika
* Iqlim tadqiqoti
* Elektrotexnika
* Moliyaviy va iqtisodiy modellashtirish
* Molekulyar modellashtirish
* Milliy mudofaa va yadro quroli
* Neft va gazni qidirish
* Kvant mexanikasi

**Parallel ishlov berish turlari**

Parallel ishlov berishning bir nechta turlari mavjud, ulardan ikkitasi eng ko’p ishlatiladigan SIMD va MIMD. SIMD yoki bitta buyruqli bir nechta ma’lumot - bu parallel ishlov berish shakli bo’lib, unda kompyuterda ikki yoki undan ortiq protsessorlar bir xil buyruqlar to’plamiga amal qiladi, har bir protsessor turli xil ma’lumotlarni qayta ishlaydi. SIMD odatda bir xil ko’rsatilgan mezonlarga asoslangan katta ma’lumotlar to’plamlarini tahlil qilish uchun ishlatiladi.

MIMD yoki bir nechta buyruqli ma’lumotlar parallel ishlov berishning yana bir keng tarqalgan shakli bo’lib, har bir kompyuterda ikkita yoki undan ortiq o’z protsessorlari mavjud va ma’lumotlarni alohida ma’lumotlar oqimlaridan oladi.

Parallel ishlov berishning boshqa kamroq qo’llaniladigan turiga MISD yoki bir nechta buyruqlar ma’lumotlari kiradi, bunda har bir protsessor bir xil kirish ma’lumotlari bilan boshqa algoritmdan foydalanadi.

Parallel ishlov berish ikki yoki undan ortiq protsessorlar yordamida bir nechta vazifalarni bajarishi mumkin bo’lgan hollarda, ketma-ket ishlov berish (shuningdek, ketma-ket ishlov berish deb ataladi) bitta protsessor yordamida bir vaqtning o’zida faqat bitta vazifani bajaradi. Agar kompyuter bir nechta tayinlangan vazifalarni bajarishi kerak bo’lsa, u bir vaqtning o’zida bitta vazifani bajaradi. Xuddi shunday, agar ketma-ket ishlov berishdan foydalanadigan kompyuter murakkab vazifani bajarishi kerak bo’lsa, u parallel protsessorga qaraganda ko’proq vaqt talab etadi.

**SMP va MMP**

Ushbu muammolarni hal qilish nosimmetrik multiprocessing tizimiga olib keldi ([SMP](https://www.techtarget.com/searchdatacenter/definition/SMP)). SMP tizimida har bir protsessor teng darajada qobiliyatli va tizim orqali ish oqimini boshqarish uchun javobgardir. Dastlab, maqsad SMP tizimlarini dasturchilarga bitta protsessor, ko’p dasturlash tizimlari kabi ko’rinishini ta’minlash edi. Biroq, muhandislar ba’zi ko’rsatmalarni noto’g’ri bajarish va dasturchilardan murakkablikni oshirishni talab qilish orqali tizimning ishlashini 10-20% oralig’ida oshirish mumkinligini aniqladilar (muammo faqat ikki yoki undan ortiq dastur bir vaqtning o’zida o’qilganda ko’rinadigan bo’lishi mumkin). va bir xil operandlarni yozing; Shunday qilib, ortib borayotgan murakkablik bilan shug’ullanish yuki faqat bir nechta dasturchilarga tushadi, keyin esa faqat juda ixtisoslashgan sharoitlarda). SMP mashinalari umumiy ma’lumotlarda o’zini qanday tutishi kerakligi haqidagi savol haligacha hal qilinmagan.

SMP tizimlarida protsessorlar soni ortishi bilan ma’lumotlarning tizimning bir qismidan boshqa barcha qismlariga tarqalish vaqti ham ortadi. Agar protsessorlar soni bir necha o’nlab oraliqda bo’lsa, tizimga qo’shimcha protsessorlarni qo’shishning unumdorligi qo’shimcha xarajatlarni oqlash uchun juda kichikdir. Uzoq tarqalish vaqtlari muammosini hal qilish uchun yuqorida aytib o’tilgan xabarlarni uzatish tizimi yaratilgan. Ushbu tizimlarda ma’lumotlarni almashuvchi dasturlar ma’lum operandlarga yangi qiymat berilganligini e’lon qilish uchun bir-biriga xabarlar yuboradi. Operandning yangi qiymatini tizimning barcha qismlariga translyatsiya qilish o’rniga, yangi qiymat faqat yangi qiymatni bilishi kerak bo’lgan dasturlarga etkaziladi. Umumiy xotira o’rniga dasturlar o’rtasida xabarlarni uzatishni qo’llab-quvvatlaydigan tarmoq mavjud. Ushbu soddalashtirish yuzlab, hatto minglab protsessorlarga bir tizimda samarali ishlash imkonini beradi. Shuning uchun bunday tizimlarga massiv parallel ishlov berish (MPP) tizimlari nomi berildi.

Eng muvaffaqiyatli MPP ilovalari katta hajmdagi ma’lumotlar bo’yicha ko’plab alohida, mustaqil operatsiyalarga bo’linishi mumkin bo’lgan muammolar uchun bo’lgan. [Ma’lumotlarni qazib olishda](https://searchsqlserver.techtarget.com/definition/data-mining) statik ma’lumotlar bazasini bir nechta qidirishni amalga oshirish zarurati mavjud . Sun’iy [intellektda](https://www.techtarget.com/searchenterpriseai/definition/AI-Artificial-Intelligence) shaxmat o’yinidagi kabi bir nechta alternativalarni tahlil qilish zarurati mavjud. Ko’pincha MPP tizimlari protsessorlar klasterlari sifatida tuzilgan. Har bir klaster ichida protsessorlar SMP tizimidagi kabi o’zaro ishlaydi. Faqat klasterlar o’rtasida xabarlar uzatiladi. Operandlarga xabarlar yoki xotira manzillari orqali murojaat qilish mumkinligi sababli, ba’zi MPP tizimlari bir xil bo’lmagan xotira manzillarini aniqlash uchun [NUMA](https://www.techtarget.com/whatis/definition/NUMA-non-uniform-memory-access) mashinalari deb ataladi.

SMP mashinalarini dasturlash nisbatan sodda; MPP mashinalari bunday emas. SMP mashinalari barcha turdagi muammolarni yaxshi hal qiladi, agar jalb qilingan ma’lumotlar miqdori unchalik katta bo’lmasa. Katta ma’lumotlar bazalarining ma’lumotlarini qazib olish kabi muayyan muammolar uchun faqat MPP tizimlari xizmat qiladi.

## 1.3 Parallel arxitekturalar

Parallel kompyuterlar tizimlari arxitekturasida ikkita jihat asosiy rol o’ynaydi:

* Protesssorlar va ularning xotiralari o’zaro qanday bog’langanligi;
* Protsessorlarning qanday o’zaro ta’sir qilishi.

Parallel algoritmlarni muhokama qilganda biz ana shu jihatlar haqida gapiramiz. Negaki u yoki bu yechimlar turli masalalar uchun turli samaradorlikka ega bo’lishi mumkin.

Kuchli va kuchsiz bog’langan mashinalar.

Kuchsiz bog’langan mashinalarda har protsessor o’z xususiy xotirasiga ega. Lekin protsessorlar o’rtasidagi aloqa tarmoq kabellari orqali amalga oshiriladi. Bu kompyuterlar klasterlarining bu kompyuterlar klasterlarining arxitekturasi shunday: klasterning har bir kompyuteri alohida kompyuter tizimi va mustaqil iashlay oladi. Parallellik bosh boshqaruvchi kompyuter orqali masalani kompyuterlarga taqsimlash hisobiga amalga oshiriladi. Tor aloqali mashinalarda barecha protsessorlar umumiy markaziy xotiradan foydalanadi. Protsessorlar o’rtasida o’zaro ta’sir shunday amalga oshiriladiki, bunda ulardan biri axborotni umumiy xotiraga yozadi, boshqalari esa shu yerdan o’qib oladi.

Protsessorlarning o’zaro ta’siri.

Kuchsiz aloqali mashinalar o’rtasida protsessorlarning o’zaro ta’siri kabellar yoki similar orqali amalga oshirilishini aytib o’tgan edik. Bunday aloqalarning tashkil qilish mumkin bo’lgan hollarini qaraymiz. Spekrning bir uchida to’liq aloqali tarmoq joylashgan. Bunda har bir protsessor boshqalari bilan ulangan. Boshqa bir uchida esa chiziqli tarmoq mavjud. Bunda protsessorlar zanjirsimon bog’langan va har bir protsessor ikkita uchidagidan tashqari ikkita qo’shnisi bilan bog’langan (uchidagi kompyuterlar bitta qo’shni bilan bog’langan). To’liq aloqali tarmoqda axborot protsessordan protsessorga juda tez uzatiladi, lekin buni tashkil qilish uchun juda uzun kabel talab qilinadi. Chiziqli tarmoqda esa axborot sekinroq uzatiladi. Sababi, axborot oraliq tugunlardan o’tadi va zanjiring biron joyidan uzilishi ma’lumtlar oqimini uzib qo’yadi. Chiziqli tarmoqqa alternative sifatida turg’unlikni oshiruvchi halqali strukturani tashkil qilish mumkin. U ham chiziqli tarmoq kabi qurilgan. Faqat zanjirning birinchi va so’nggi zvenosi o’zaro ulangan bunday tarmoqda axborot tez tarqaladi

Parallel arxitekturada protsessorlarni ulashning bundan boshqa imkoniyatlari ham mavjud. Bularga daraxtsimon tarmoqlar (protsessorlar bironta daraxtni hosil qiladi) va ikki o’lchovli chambarani umumlashtiradigan giperkublarni misol qilish mumkin.

* Parallel algoritmlar analizining prinsiplari
* Parallel algoritmlar bilan ishlashda bizni ikki tushuncha qiziqtiradi:
* Tezlanish koeffitsiyenti
* Qiymati

Parallel algoritmlarning tezlanish koeffitsiyenti optimal ketma-ket algoritmning qanchalik tez ishlashini ko’rsatadi. Ma’lumki optimal tartiblash algoritmi O(Nlog N) ta operatsiyani talab qiladi. O(N) murakkablikdagi parallel tartiblash algoritmining tezlanish koeffitsiyenti O(log N) ni tashkil qiladi.

Bizni qiziqtiadigan ikkinchi tushuncha – parallel algoritmning qiymati bo’lib, uni biz ishlatilayotgan protsessorlar sonining algoritm murakkabligiga ko’paytmasi orqali aniqlaymiz. Agar bizning holatda parallel tartiblash algoritmi O(N) ta operatsiya uchun kiruvchi yozuvlar soniga teng protsessorlarni talab qilsa, uning qiymati O(N^) ga teng. Bu parallel tartiblash algoritmi qimmatliroq ekanligini bildiradi, ya’ni bitta protsessordagi ketma-ket tartiblash algoritmining qiymati uning murakkabligi bilan mos keladi va O(Nlog N) ga teng.

Zarur tushunchalardan yana biri vazifaning tarkibiy qismlarga ajralishidir. Agar bizning parallel tartiblash algoritmimiz uchun yagona imkoniyat protsessorlar sonining kiruvchi yozuvlar soniga tengligi shart bo’lsa, unda bunday algoritm kiruvchi yozuvlar sonining yetarlicha katta bo’lishi bilan befoyda bo’lib qoladi. Ketma-ket tartiblash algoritmida hech qanday o’xshash cheklovlar yo’q. Bizni ko’proq protsessorlar soni kiruvchi ma’lumotlar potensial hajmidan yetarlicha kam bo’lgan va bu son kirish uzunligining ortishi bilan kattalashishni talab qilmaydigan parallel tartiblash algoritmlari qiziqtiradi.

PRAM modeli

Parallel sistemalaning muammolaridan biri ma’lumotlarni xotiradan o’qish va xotiraga yozishdir. Masalan, agar ikkita protsessor ma’lumotlarni umumiy xotiraning aynan bitta joyiga yozishga urinsa nima bo’ladi?

Ilgari biz ko’rgan algoritmlarda bu algoritmni amalga oshiradigan mashina ilgaridan berilgan xotira yacheykasi (RAM) ga to’g’ridan-to’g’ri kirish imkoniyatiga ega. Hozir biz qaraydigan algoritmlar bunday mashinalarning parallel varianti (PRAM) ga asoslangan. Bizning PRAM mashinalarning protsessorlari o’zaro chambarchas bog’langan va umumiy xotira blokidan foydalanadi. Har bir protsessorda uncha katta hajmda bo’lmagan ma’lumotlarni saqlash imkoniyatiga ega bo’lgan bir nechta registrlar mavjud, ma’lumotlarning asosiy qismi esa umumiy xotirada saqlanadi.

Protsessorlar orasidagi chambarchas bog’liqlikdan tashqari biz yana ular hammasi xotiradan ma’lumotlarni o’qish, ular ustida operatsiya bajarish va natijani xotiraga yozishdan iborat bo’lgan bir xil siklni amalga oshiradi deb faraz qilamiz. Bu barcha protsessorlar bir vaqtda xotirani o’qishini, bir vaqtda o’qilgan ma’lumotlarni qayta ishlashini, bir vaqtda yozuvni ham bajarishini anglatadi. Xotira yacheykalari ustidagi bahs nafaqat ma’lumotlarni o’qishda, balki natijani yozishda ham kelib chiqadi. Uch qadamli siklning shartlari agar Y protsessor xotira yacheykasidagi ma’lumotlarni o’zgartirgan vaqtda, X protsessor hozirgina o’qilgan ma’lumotlarni qayta ishlashi natijasida biz nima bo’lishi haqida qayg’urmasak ham bo’lishini anglatadi. Bundan tashqari, bitta protsessor xotiradan ma’lumotlarni o’qiyotgan vaqtda, ikkinchisi unga biron ma’lumot yozishga urinishi kabi vaziyatlar ham yuzaga kelmaydi.

Bahslarga faqatgina xotiraga kirishga raqobatli yoki maxsus huquq berish orqali ruxsat berish mumkin. Raqobatli kirishda xotiraning aynan bitta yacheykasiga bir vaqtda bir nechta protsessor murojaat qilishi mumkin. Maxsus kirishda esa berilgan xotira yacheykasiga aniq momentda faqat bittagina protsessor murojaat qila oladi, bir vaqtdagi ikkita murojaat qilishga urinish esa xatolik haqidagi xabarning paydo bo’lishiga olib keladi. Raqobatli kirish o’qish vaqtida muammo keltirib chiqarmaydi. Bundan tashqari bizga maxsus o’qish huquqi bilan ishlaydigan algoritmlar ham kerak. Maxsus o’qish huquqiga ega bo’lgan bir nechta protsessor bir vaqtda bitta xotira yacheykasiga murojaat qilsa xatolik paydo bo’ladi.

Bundan tashqari yozuv vaqtida maxsus va raqobatli kirishlarni tanlashda ham muammo mavjud. Maxsus kirishda har qaysi xotira yacheykasiga yozuv huquqi faqat bitta protsessorga beriladi, bir necha protsessorlar yozishga harakat qilsa, xatolik paydo bo’ladi. Lekin ikkita protsessor ikkita xotira yacheykasiga bir vaqtda yoza oladi. Raqobatli kirishda esa, masala birmuncha murakkab, ya’ni kelib chiqadigan konfliktlarga ruxsat bera olish kerak. Darajaga ega modelda har bir protsessorga daraja beriladi va yozuv huquqi kattaroq darajali protsessorga beriladi.

Modelning soddalashtirilgan protsessor darajasi uning nomeriga to’g’ri keladi, ya’ni nomer qancha kichik bo’lsa, daraja shuncha katta bo’ladi. Agar, masalan, bitta xotira yacheykasiga 4- va 7- nomerli protsessorlar yozishga urinsa, huquq 4-nomerli protsessorga beriladi.

Ixtiyoriy kirishli modelda raqobat qiladigan protsessorlardan ixtiyoriy biri olinishi mumkin. Oddiy modelda esa faqatgina yoziluvchi ma’lumotlar ustma-ust tushgandagina bir vaqtda yozishga ruxsat beriladi. Kombinatsiyali modelda yoziluvchi ma’lumotlar ustidasistema bir qancha amallarni bajaradi. Masalan, ularning yig’indisi, ko’paytmasi, eng katta yoki eng kichik elementi yoki mantiqiy operatsiya natijasi (va, yoki) yozilgan bo’lishi mumkin. Turli holatlarda bu imkoniyatlardan har biri foydali bo’lishi mumkin.

Bundan kelib chiqadiki, bizda yozish va o’qish imkoniyatlarining 4 xil kombinatsiyasi mavjud:

* Raqobatli o’qish / raqobatli yozish (CRCW);
* Raqobatli o’qish / maxsus yozish (CREW);
* Maxsus o’qish / raqobatli yozish (ERCW);
* Maxsus o’qish / maxsus yozish (EREW);

## 1.4 Parallel hisoblashning afzallilklari va kamchiliklari.

Afzalliklar

* Parallel hisoblash vaqtni tejaydi, bu esa ilovalarni devor soatlarida qisqaroq vaqt ichida bajarishga imkon beradi.
* Qisqa vaqt ichida kattaroq muammolarni hal qiling.
* Seriyali hisoblash bilan solishtirganda, parallel hisoblash murakkab, real dunyo hodisalarini modellashtirish, simulyatsiya qilish va tushunish uchun ancha mos keladi.
* Biror vazifani bajarish uchun ko'proq resurslarni sarflash, potentsial xarajatlarni tejash bilan yakunlash vaqtini qisqartiradi. Parallel kompyuterlar arzon, tovar komponentlardan tuzilishi mumkin.
* Ko'pgina muammolar juda katta va/yoki murakkab bo'lib, ularni bitta kompyuterda, ayniqsa cheklangan kompyuter xotirasida hal qilish amaliy yoki imkonsizdir.
* Bir nechta hisoblash resurslaridan foydalangan holda bir vaqtning o'zida ko'p narsalarni qilishingiz mumkin.
* Keng tarmoqli (WAN) yoki hatto internetda kompyuter resurslaridan foydalanishi mumkin.
* Bu sizni tartibli saqlashga yordam beradi. Agar sizda Internet mavjud bo'lsa, u holda muloqot va ijtimoiy tarmoq osonlashtirilishi mumkin.
* U katta ma'lumotlarni saqlash va tezkor ma'lumotlarni hisoblash imkoniyatiga ega.

Kamchiliklari

* Parallel arxitekturani maqsadli dasturlash biroz qiyin, ammo to'g'ri tushunish va amaliyot bilan siz borishingiz mumkin.
* Parallel hisoblashdan foydalanish ko'p yadroli protsessorlar yordamida hisoblash va ma'lumotlarni ko'p talab qiladigan muammolarni hal qilishga imkon beradi, lekin ba'zida bu bizning ba'zi boshqarish algoritmimizga ta'sir qiladi va yaxshi natijalar bermaydi va bu parallel variant tufayli tizimning konvergentsiyasiga ham ta'sir qilishi mumkin. .
* Qo'shimcha xarajat (ya'ni, bajarilish vaqtining ko'payishi) ma'lumotlarni uzatish, sinxronizatsiya, aloqa, ipni yaratish/yo'q qilish va hokazolar bilan bog'liq. Bu xarajatlar ba'zan juda katta bo'lishi mumkin va aslida parallellashtirish natijasida olingan daromaddan oshib ketishi mumkin.
* Ishlashni yaxshilash uchun turli maqsadli arxitekturalar uchun turli xil kodlarni sozlash kerak.
* Klasterlar bo'lsa, yaxshiroq sovutish texnologiyalari talab qilinadi.
* Ko'p yadroli arxitekturada quvvat sarfi juda katta.
* Parallel echimlarni amalga oshirish qiyinroq, ularni disk raskadrovka qilish yoki to'g'riligini isbotlash qiyinroq va ular ko'pincha aloqa va muvofiqlashtirish xarajatlari tufayli seriyali hamkasblariga qaraganda yomonroq ishlaydi.

## 1.5 Adabiyotlar tahlili va metodologiya.

Dunyoda uzluksiz algoritmlar rivojining zamonaviy bosqichida murakkab jarayonlarni tez va sifatli tahlil qilish va ulkan hajmdagi axborotni real vaqtda qayta ishlash tezligini oshirish usullari va algoritmlarini ishlab chiqish hamda uzluksiz ishlov berish texnologiyalari asosida zamonaviy protsessorlarning hisoblash jarayonlarini tezkorligini oshirish masalalari o'rganilmoqda. Hisoblash jarayonidagi buyruqlarni konveyrli va uzluksiz shakllantirish, katta hajmdagi buyruqlarni amalga oshirishning yangi texnologiyalarini yaratish asosida tezkorlikning sifatli o'sish ko'rsatkichiga erishish mumkin. SHu kabi tezkor hisoblash usul va algoritmlarni takomillashtirish va dasturiy vositalarni ishlab chiqish dolzarb masalalardan biri hisoblanadi.

Men dissertatsiyamni qilish jarayonida Andijon qishloq xo‘jaligi va agrotexnologiyalar instituti, katta o‘qituvchisi Odiljon Qobulovich Abduraxmonovning “Ko‘p yadroli protsessorda kubik bazisli splaynlar asosida parallel algoritmlarni amalga oshirish tuzilmasini ishlab chiqish” nomli dissertatsiyasini o’rganib chiqmoqdaman. Muallif bu dissertatsiyada signallarga raqamli ishlov berish masalalarini yechishda parallel va uzluksiz algoritmlar yaratish va ishlov berish jarayonlarini ko‘p yadroli arxitektura asosida amalga oshirishga qaratilgan muammolarning ba’zi masalalarini o‘rganib chiqqan.

Respublikamizda ham signallarga raqamli ishlov berish masalalarini yechishda uzluksiz algoritmlar yaratish va ishlov berish jarayonlarini ko'p yadroli arxitektura asosida amalga oshirishga qaratilgan ilmiy tadqiqot ishlari olib borilmoqda.

Mazkur vazifalarni amalga oshirishda, jumladan, signallarga raqamli ishlov berishda kubik splaynlar asosida ko'p yadroli protsessorlar uchun uzluksiz algoritmlar yaratish, ma'lumotlarga tezkor ishlov berish, signallarni raqamli ishlash algoritmlarining samaradorligini oshirish va signallarning lokal xususiyatlarini aniqlashga mo'ljallangan usul, algoritm, apparat va dasturiy vositalarni ishlab chiqish muhim vazifalardan biri hisoblanadi.

O'zbekiston Respublikasi Prezidentining 2017 yil 7 fevraldagi PF-4947-son «O'zbekiston Respublikasini yanada rivojlantirish bo'yicha Harakatlar strategiyasi to'g'risida» gi va 2018 yil 19 fevraldagi PF-5349-son «Axborot texnologiyalari va kommunikatsiyalari sohasini yanada takomillashtirish chora-tadbirlari to'g'risida» gi Farmonlari,2018 yil 7 martdagi Vazirlar Mahkamasining «Aloqa, axborotlashtirish va telekommunikatsiya xizmatlari sifatini yanada yaxshilashga doir chora-tadbirlar to'g'risida» gi 185-sonli qarori hamda, mazkur faoliyatga tegishli boshqa me'yoriy-huquqiy hujjatlarda belgilangan vazifalarni amalga oshirishga ushbu maqola tadqiqoti ma'lum darajada xizmat qiladi.

Yana bir tadqiqotchilardan biri Toshkent Axborot Texnologiyalari Universitetining Katta o‘qituvchisi Mallayev Oybek Usmonqulovichdir. Uning “Klasterdagi ko‘p yadroli protsessorlarda kubik splayn qurish uchun parallel algoritm” mavzusidagi maqolasini o‘rganib chiqmoqdaman. Bu maqolada kubik spline yordamida ma‘lumotlarni parallel siqishni hisoblash imkoniyati o‘rganilgan, seysmik signallarni raqamli qayta ishlash jarayonini parallel qilish yo‘llari ko‘rib chiqilgan va parallel algoritmlarning asosiy ishlash ko‘rsatkichlari ketma-ket algoritmlar bilan taqqoslangan.