

Qiskit, Microsoft Q#, Cirq, PyQuil VA IBM Quantum Composer-ga E'TIBOR QARATGAN HOLDA KVANT HISOBLASHDA DASTURLASH TILLARINING AHAMIYATI

Norqobilov Sobir Hamza o'g'li

Termiz davlat universiteti o'qituvchisi. O'zbekiston

Annotatsiya: Ushbu maqola Qiskit, Microsoft Q#, Cirq, PyQuil va IBM Quantum Composer-ga e'tibor qaratgan holda kvant hisoblashda dasturlash tillarining ahamiyatini o'rganib chiqilgan. Unda ularning o'ziga xos xususiyatlari va imkoniyatlari muhokama qilinadi, shuningdek, yangi boshlanuvchilar uchun ko'rsatmalar va manbalar taqdim etildi. Maqolada sohadagi muammolar va olib borilayotgan izlanishlarga ham to'xtalib o'tildi. Kvant hisoblashlari rivojlanishda davom etar ekan, dasturlash tillari ushbu kuchli tizimlarning imkoniyatlarini ochishda muhim rol o'ynaydi.

Kalit so'zlar: Qiskit, Microsoft Q#, Cirq, PyQuil, IBM Quantum Composer, dasturlash tillari, Kvant hisoblash, klassik kompyuterlar, Klassik mexanikada, Interference.

Аннотация: В этой статье рассматривается важность языков программирования в квантовых вычислениях с упором на Qiskit, Microsoft Q#, Cirq, PyQuil и IBM Quantum Composer. В нем обсуждаются их уникальные функции и возможности, а также предоставляются инструкции и ресурсы для начинающих. В статье также затронуты проблемы в данной области и проводимые исследования. Поскольку квантовые вычисления продолжают развиваться, языки программирования будут играть важную роль в раскрытии потенциала этих мощных систем.

Ключевые слова: Qiskit, Microsoft Q#, Cirq, PyQuil, IBM Quantum Composer, языки программирования, квантовые вычисления, классические компьютеры, классическая механика, интерференция.

Annotation: This article explores the importance of programming languages in quantum computing, focusing on Qiskit, Microsoft Q#, Cirq, PyQuil, and IBM Quantum Composer. It discusses their unique features and capabilities, as well as providing instructions and resources for beginners. The article also touched upon the problems in the field and the ongoing research. As quantum computing continues to evolve, programming languages will play an important role in unlocking the potential of these powerful systems.

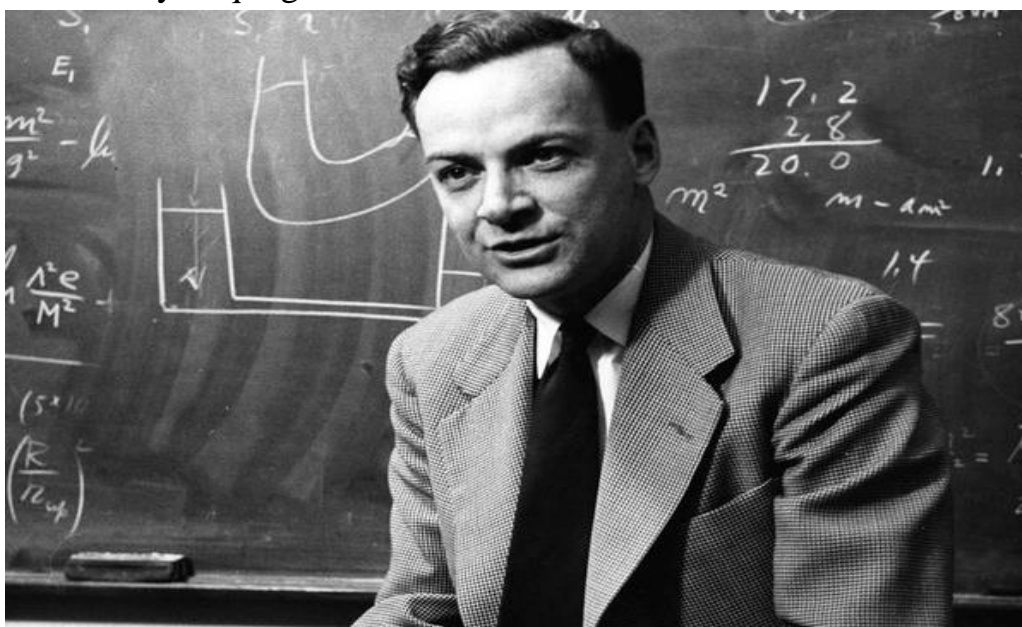
Key words: Qiskit, Microsoft Q#, Cirq, PyQuil, IBM Quantum Composer, programming languages, Quantum computing, classical computers, Classical mechanics, Interference.

Kirish: Kvant hisoblash - tez rivojlanayotgan soha bo'lib, klassik kompyuterlar bilan hal qilib bo'lmaydigan muammolarni hal qilish uchun kvant mexanikasi kuchidan foydalanishga intiladi. 0 yoki 1 ni ifodalash uchun bitlardan foydalanadigan klassik kompyuterlardan farqli o'laroq, kvant kompyuterlari bir vaqtning o'zida ikkala holatning superpozitsiyasida mavjud bo'lishi mumkin bo'lgan kvant bitlari yoki qubitlardan foydalanadi. Bu kvant kompyuterlariga ko'plab hisob-kitoblarni parallel ravishda bajarishga imkon beradi, bu ularni katta sonlarni faktoring qilish, murakkab kimyoviy reaksiyalarni simulyatsiya qilish va logistikani optimallashtirish kabi vazifalar uchun juda mos keladi.

Kvant hisoblashning ulkan salohiyatiga qaramay, klassik kompyuterlardan ustun turadigan keng ko'lamli kvant kompyuterini yaratish hali ham katta muammo bo'lib qolmoqda. Biroq, butun dunyo bo'ylab tadqiqotchilar va muhandislar ularning kuchidan foydalanish uchun kvant kompyuterlari va kvant algoritmlarini ishlab chiqish ustida ishlamoqda.

Adabiyotlarning tahlili. Hisoblash uchun kvant mexanikasidan foydalanish g'oyasi 1980-yillarda fizik Richard Feynman kvant tizimlarini klassik kompyuterlarga qaraganda samaraliroq simulyatsiya qilish uchun kvant kompyuteri g'oyasini taklif qilgan paytdan boshlanadi. 1994 yilda matematik Peter Shor kriptografiyaga sezilarli ta'sir ko'rsatadigan eng mashhur klassik algoritmdan ko'ra ko'p sonlarni eksponent ravishda tezroq faktorlashi mumkin bo'lgan kvant algoritmini ishlab chiqdi.

Dastlabki yillarda kvant hisoblash asosan nazariy edi va 1990-yillarning oxiriga kelib kichik hajmdagi kvant hisoblashning birinchi eksperimental namoyishlari amalga oshirildi. O'shandan beri kvant hardware, dasturiy ta'minot va algoritmlarni ishlab chiqishda sezilarli yutuqlarga erishildi.



1-rasm. Rasmda Richard Feynman

Kvant kompyuterining ishlashi atom va subatomik miqyosdagi zarrachalarning harakatini tavsiflovchi kvant mexanikasining asosiy tamoyillariga asoslanadi. Kvant hisoblashning asosi bo'lgan bir nechta asosiy printsiplar mavjud, ular orasida superpozitsiya, entanglement(chalkashlik) va interferensiya mavjud.

Metodologiya. Dasturlash tillari kvant hisoblashlari uchun zarurdir, chunki ular kvant tizimlarining xatti-harakatlarini boshqarish va kvant algoritmlarini loyihalash vositalarini ta'minlaydi. Kvant mexanikasining kvant hisoblashlari asosidagi tamoyillari klassik hisoblashlardan juda farq qiladi, shuning uchun kvant hisoblashlari uchun dasturlash tillari ushbu tamoyillarni hisobga olish uchun moslashtirilishi kerak. Ular ishlab chiquvchilarga qubitlarni manipulyatsiya qilish va kvant algoritmlarini bajarish uchun kvant sxemalarini yaratish bo'yicha ko'rsatmalar yozishga imkon beradi. Ushbu maqolada biz kvant hisoblash uchun mavjud bo'lgan turli xil dasturlash tillarini va ular ushbu inqilobiy texnologiyaning imkoniyatlarini ochish uchun qanday ishlatilishini o'rganamiz.

Kvant hisoblashlarini tushunish uchun kvant mexanikasi tamoyillari haqida asosiy tushunchaga ega bo'lish muhimdir.

Klassik mexanikada zarralar aniq belgilangan pozitsiya va momentga ega bo'lishi mumkin va ularning xususiyatlarini klassik o'zgaruvchilar bilan tavsiflash mumkin. Bundan farqli o'laroq, kvant mexanikasida zarralar to'liq funktsiyalari bilan tavsiflanadi, bu esa zarrachani ma'lum bir holatda topish ehtimolini beradi.

Tahlil va natijalar. Qubit - bu 0 yoki 1 holatda bo'lishi mumkin bo'lgan klassik bitga o'xshash ikki holatli kvant tizimi. Biroq, qubitlar bir vaqtning o'zida ikkala holatning superpozitsiyasida mavjud bo'lishi mumkin, ya'ni bitta qubit klassik bitga qaraganda ko'proq holatlarni ifodalashi mumkin. Bundan tashqari, qubitlar chigal bo'lishi mumkin, ya'ni bir qubitning holati boshqasining holatiga bog'liq bo'lishi mumkin, hatto ular katta masofalar bilan ajratilgan bo'lsa ham.

Klassik va kvant hisoblash o'rtasidagi yana bir asosiy farq bu xatolarning tabiatidir. Kvant hisoblashda xatolar qubitlar va boshqa omillar orasidagi kvant kogerentligini yo'qotish bo'lgan dekoherentlik tufayli yuzaga kelishi mumkin. Kvant kompyuterlari uchun xatolarni tuzatishning samarali strategiyalarini ishlab chiqish doimiy tadqiqot sohasidir.

Kvant gatelari - bu bir yoki bir nechta qubitlarda o'z holatlarini boshqarish uchun harakat qiladigan operatsiyalar. Ular klassik hisoblashda AND, OR va NOT gatelari kabi logical gatelarga o'xshash va turli kvant hisoblarini bajarish uchun ishlatilishi mumkin.

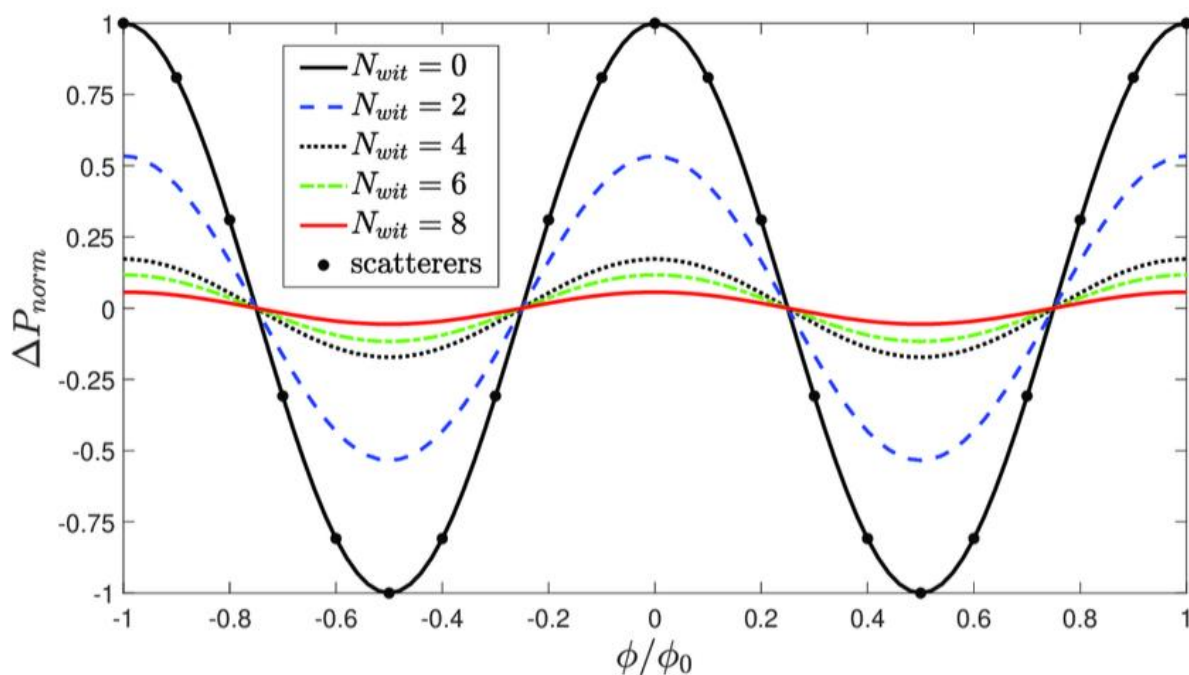
Kvant sxemalari kvant quantum gatelaridan iborat bo'lib, kvant kompyuterida bajariladigan operatsiyalar ketma-ketligini ifodalaydi. Ulardan kvant algoritmlarini

amalga oshirish uchun foydalanish mumkin, ya'ni kvant mexanikasining o'ziga xos xususiyatlaridan muayyan muammolarni klassik algoritmlarga qaraganda samaraliroq hal qilish uchun foydalanadigan operatsiyalar ketma-ketligi.

Superpozitsiya kvant mexanikasining asosiy printsipti bo'lib, qubitlarning bir vaqtning o'zida bir nechta holatda bo'lishiga imkon beradi. Klassik hisoblashda bit 0 yoki 1 bo'lishi mumkin, ammo kvant hisoblashda qubit bir vaqtning o'zida ikkala holatning superpozitsiyasida bo'lishi mumkin. Bu shuni anglatadiki, bitta qubit klassik bitga qaraganda ko'proq holatlarni ifodalashi mumkin va bir nechta qubitlar bir vaqtning o'zida juda ko'p holatlarni ifodalashi mumkin.

Entanglement kvant mexanikasining yana bir asosiy printsipti bo'lib, u ikki yoki undan ortiq qubitlarni bir-biridan katta masofalar bilan ajratilgan bo'lsa ham, bir qubitning holati boshqasining holatiga bog'liq bo'ladigan tarzda correlate qilishga imkon beradi. Bu shuni anglatadiki, bir qubitning holatini o'lchash ular orasidagi masofadan qat'i nazar, boshqa qubitning holatini bir zumda aniqlashi mumkin.

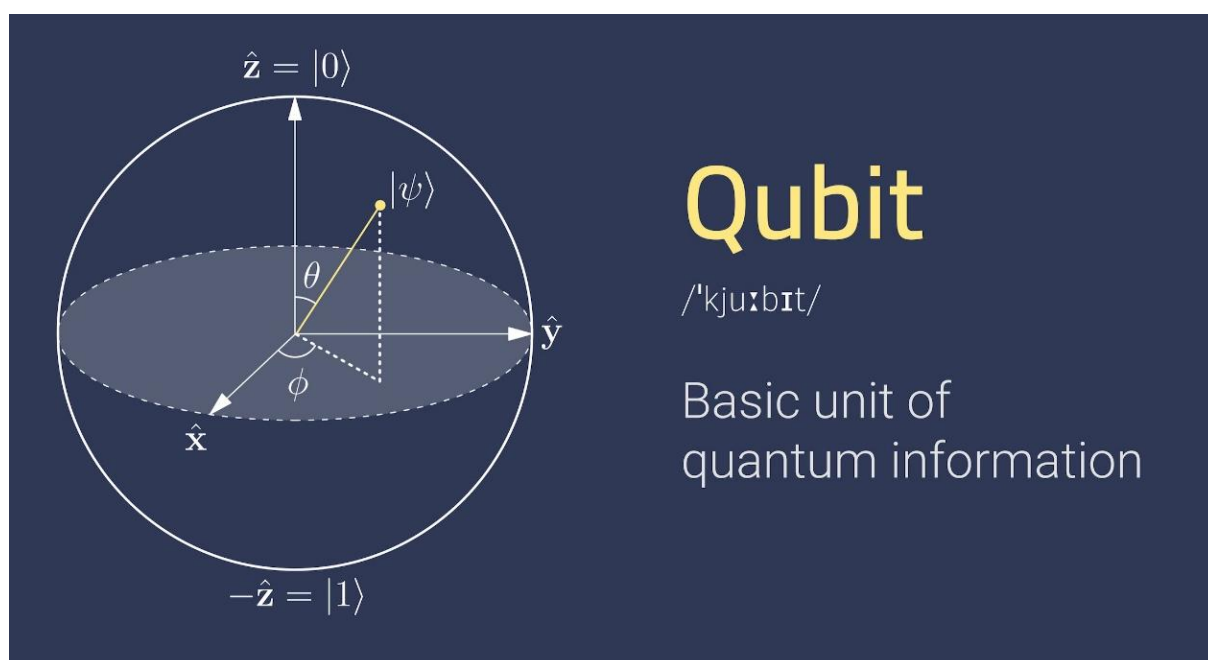
Interference (Interferentsiya) kvant holatlarining o'zaro ta'sirini tavsiflovchi printsiptir. Ikki yoki undan ortiq kvant holatlari o'zaro ta'sirlashganda, ular bir-biriga xalaqit berishi mumkin, natijada constructive yoki destructive interferentsiya paydo bo'ladi. Bu interfering holatlarning fazasiga qarab, ma'lum bir holatni o'lchash ehtimolini kuchaytirish yoki kamaytirish uchun ishlatilishi mumkin.



2-rasm. Kvant mexanikasi

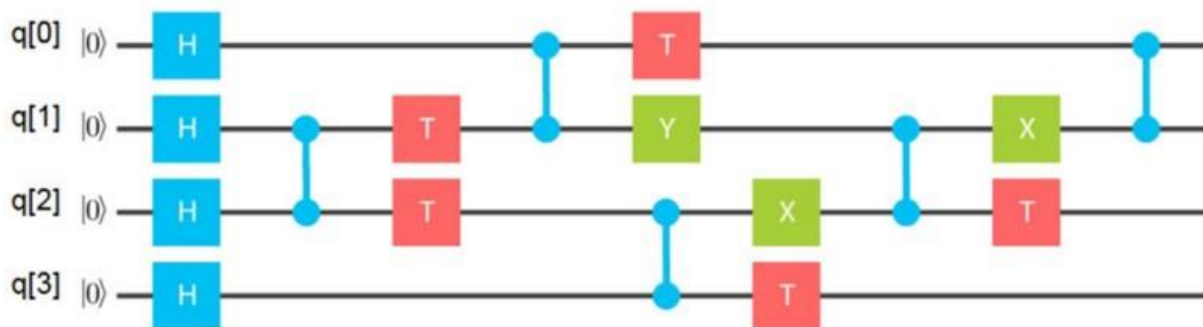
Kvant mexanikasining ushbu tamoyillari klassik hisoblashda logic gatelarga o'xshash kvant gatelari yordamida qubitlarning holatini boshqarish uchun ishlatiladi. Kvant gatelari yordamida qubitlarning holatini manipulyatsiya qilish orqali kvant algoritmlarini amalga oshirish uchun kvant sxemalarini qurish mumkin, ular klassik algoritmlarga qaraganda aniq muammolarni hal qilish uchun ishlatilishi mumkin.

Qubitlar kvant kompyuterining asosiy qurilish bloklaridir. Ular klassik hisoblashda ishlatiladigan klassik bitlarga o'xshash, ammo ular kvant mexanikasi tamoyillariga muvofiq ishlaydi. Qubit ikkita holat, 0 va 1 superpozitsiyasida bo'lishi mumkin va boshqa qubitlar bilan ham aralashib ketishi mumkin. Qubitning holatini klassik bitni olish uchun o'lchash mumkin, ammo kubitni o'lchash harakati uning superpozitsiya holatini buzadi.



3-rasm. Quantum gatelar

Quantum gatelar - bu qubitlarning holatini boshqarish uchun ishlatiladigan operatsiyalar. Ular klassik hisoblashda qo'llaniladigan logic gatelarga o'xshash, ammo ular kvant mexanikasi tamoyillariga muvofiq ishlaydi. Eng keng tarqalgan quantum gatelari - Hadamard gate. Pauli gatelar, phase gate, va CNOT gatelar.



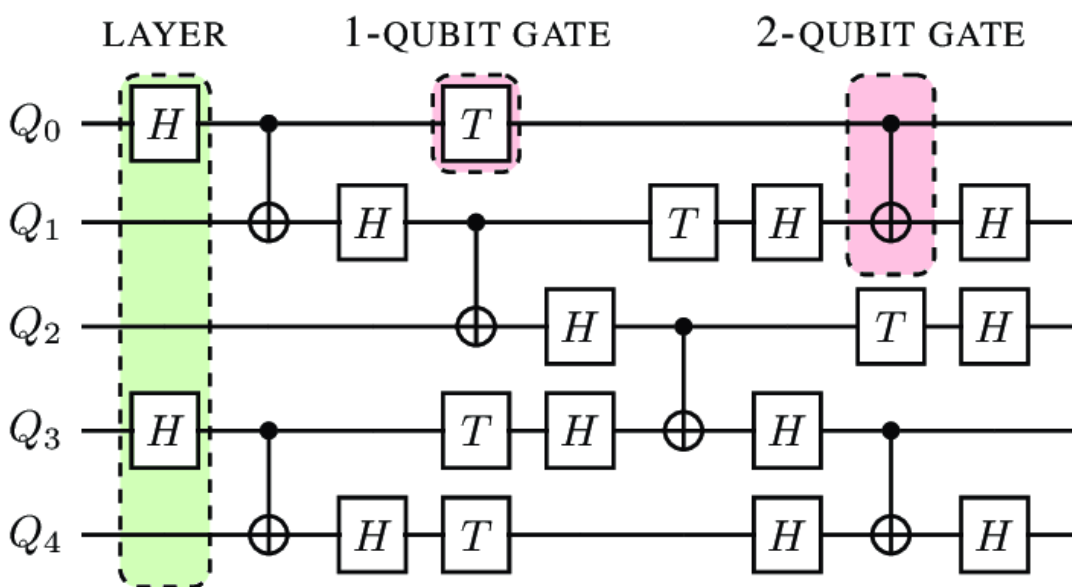
(a) A 4x1 universal random quantum circuit



(b) The "bristle-brush" pattern formed by the gates applied to qubit q[1]

4-rasm. Quantum circuit - Kvant sxemalari

Kvant sxemalari kvant algoritmlarini implement qilish uchun ishlatiladigan quantum gatelarini tartibga solishdir. Kvant sxemasi bir yoki bir nechta qubitlarga o‘z holatini o‘zgartirish uchun harakat qiladigan bir qator quantum gatelaridan iborat.



5-rasm. Quantum circuit - Kvant sxemalari

Kvant hisoblash qo‘llaniladigan sohalar

Kvant hisoblash ilm-fan, muhandislik va sanoatning ko‘plab sohalarini inqilob qilish imkoniyatiga ega va u allaqachon keng ko‘lamli ilovalar uchun o‘rganilmoqda. Kvant hisoblashning hozirgi va potentsial ilovalaridan ba’zilari:

Kriptografiya

Kvant hisoblash RSA va elliptic curve kriptografiya kabi raqamli aloqalarni himoya qilish uchun ishlatiladigan ko‘plab kriptografik tizimlarni buzish potentsialiga ega. Biroq, u kvant kalitlarini taqsimlash kabi kvant hujumlariga chidamli yangi kriptografik tizimlarni ishlab chiqish uchun ham ishlatilishi mumkin.

Optimallashtirish

Rejalashtirish, marshrutlash va resurslarni taqsimlash kabi ko‘plab real muammolarni klassik algoritmlar yordamida hal qilish qiyin. Kvant hisoblash klassik algoritmlarga qaraganda samaraliroq bo‘lgan yangi optimallashtirish algoritmlarini ishlab chiqish uchun ishlatilishi mumkin, bu esa ushbu muammolarni tezroq va aniqroq hal qilish imkonini beradi.

Simulyatsiya

Kvant mexanikasi klassik kompyuterlar yordamida, hatto nisbatan kichik tizimlar uchun ham taqlid qilish juda qiyin. Kvant hisoblashlari kvant tizimlarini yanada aniqroq va samaraliroq simulyatsiya qilish uchun ishlatilishi mumkin, bu materialshunoslik, kimyo va boshqa sohalarida yangi kashfiyotlar qilish imkonini beradi.

Machine learning

Kvant hisoblash klassik algoritmlarga qaraganda tezroq va aniqroq bo‘lgan yangi machine learning algoritmlarini ishlab chiqish uchun ishlatilishi mumkin, bu ma’lumotlarni tahlil qilish, tasvir va nutqni aniqlash va boshqalarda yangi ilovalarga imkon beradi.

Moliyaviy modellashtirish

Opsion narxlari va risklarni tahlil qilish kabi ko‘plab moliyaviy modellar klassik kompyuterlar yordamida amalga oshirish qiyin bo‘lgan keng ko‘lamli simulyatsiyalarni talab qiladi. Kvant hisoblashlari yanada aniqroq va samarali moliyaviy modellarni ishlab chiqish uchun ishlatilishi mumkin, bu esa xatarlarni yaxshiroq boshqarish va investitsiya qarorlarini qabul qilish imkonini beradi.

Dori kashfiyoti

Kvant hisoblash usuli molekular va oqsillarning xatti-harakatlarini klassik kompyuterlarga qaraganda aniqroq va samaraliroq simulyatsiya qilish uchun ishlatilishi mumkin, bu esa dori-darmonlarni tezroq va aniqroq topish va ishlab chiqish imkonini beradi.

Bular kvant hisoblashning joriy va potentsial qo'llanilishiga bir nechta misollar. Texnologiya rivojlanishda va yetuklikda davom etar ekan, u fan, texnika va sanoatning ko'plab sohalariga yanada katta ta'sir ko'rsatishi mumkin.

Kvant hisoblashning asosiy qiyinchiliklari va cheklovlari.

Kvant hisoblash ko'plab sohalarda inqilob qilish potentsialiga ega bo'lsa-da, u bir qator qiyinchiliklar va cheklovlarga ham duch keladi. Kvant hisoblashning asosiy qiyinchiliklari va cheklovlaridan ba'zilari:

Xatolarni tuzatish

Kvant hisoblash tizimdagi xatolar va noiselarga juda sezgir. Qubitlar juda nozik va muvofiqlikni osongina yo'qotishi mumkinligi sababli, kvant kompyuterlarining ishonchli ishlashi uchun xatolarni tuzatish juda muhimdir. Biroq, kvant tizimida xatolarni tuzatishni amalga oshirish qo'shimcha qubitlar va murakkab algoritmlarni talab qiladigan qiyin vazifadir.

Scalability(Masshtablilik)

Kvant hisoblashning asosiy afzalliklaridan biri klassik kompyuterlarga nisbatan eksponensial tezlikni oshirish imkoniyatidir. Biroq, bu tezlikka erishish uchun ko'p sonli qubitlar kerak bo'ladi. Hozirgi vaqtda kvant kompyuterlari o'nlab yoki yuzlab qubitlar bilan cheklangan, bu ko'plab amaliy muammolarni hal qilish uchun yetarli emas. Kvant tizimlarini minglab yoki millionlab qubitlarga kengaytirish katta texnik muammodir.

Noise va dekogerentlik

Kvant tizimlari environmental noisega va dekogerentlikka juda sezgir. Kvant tizimi va atrof-muhit o'rtasidagi har qanday o'zaro ta'sir tizimning uyg'unligini yo'qotishiga va kvant holatini yo'q qilishga olib kelishi mumkin. Bu uzoq vaqt davomida qubitlarning nozik kvant holatini saqlab qolishni qiyinlashtiradi.

Dasturlashning murakkabligi

Kvant kompyuterini dasturlash klassik dasturlashdan juda farq qiladigan yangi ko'nikmalar va vositalar to'plamini talab qiladi. Kvant algoritmlarini ishlab chiqish va ularni kvant zanjirlariga o'tkazish kvant mexanikasi va kompyuter fanlari bo'yicha tajriba talab qiladigan murakkab va qiyin vazifadir.

Hardware cheklovlari

Hozirgi kvant kompyuterlari o'ta o'tkazuvchan qubitlar yoki tutilgan ionlar kabi mavjud hardware texnologiyalari bilan cheklangan. Har bir texnologiyaning o'ziga xos afzalliklari va cheklovlari bor va oxir-oqibat qaysi texnologiya eng muvaffaqiyatli bo'lishi aniq emas.

Narxi

Kvant kompyuterini yaratish va ishlatish qimmat va murakkab jarayondir. Kvant uskunasi yuqori narxi va ixtisoslashtirilgan tajribaga bo'lgan ehtiyoj kvant hisoblash imkoniyatlarini faqat bir nechta tadqiqot institutlari va kompaniyalari bilan cheklaydi.

Qisqa qilib aytadigan bo'lsak, kvant hisoblash bir qator qiyinchiliklar va cheklovlarga duch keladi, jumladan xatolarni tuzatish, Scalability, noise va dekoherentlik, dasturlashning murakkabligi, hardware cheklovlari va xarajatlar. Ushbu muammolarni hal qilishda muvaffaqiyatga erishilgan bo'lsa-da, bu soha hali ham dastlabki bosqichda va kvant hisoblashning to'liq imkoniyatlarini amalga oshirish uchun ko'proq tadqiqotlar talab etiladi.

Kvant kompyuterlari dasturlash tillari

Kvant hisoblash sohasi ishlabchiquvchilar uchun mavjud bo'lgan bir nechta dasturlash tillariga ega. Bu tillar dasturchilarga kvant algoritmlarini yozish va ularni kvant kompyuterlarida ishga tushirishga yordam berish uchun yaratilgan. Kvant hisoblash uchun mashhur dasturlash tillaridan ba'zilari:

Qiskit - IBM tomonidan ishlab chiqilgan Python-ga asoslangan kvant hisoblash tizimi. U kvant sxemalarini yaratish va ishga tushirish, kvant kompyuterlarini taqlid qilish va IBM kvant hardware si bilan aloqa o'rnatish uchun toollar to'plamini taqdim etadi. Qiskit turli kvant operatsiyalarini qo'llab-quvvatlaydi, masalan, bitta qubit va ko'p qubit operatsiyalari uchun gatellar, shuningdek, o'lchovlar va noiselarni modellashtirish. Shuningdek, u gibridd kvant-klassik algoritmlar uchun klassik hisoblashni qo'llab-quvvatlaydi.

Microsoft Q# — Microsoft tomonidan ishlab chiqilgan kvant hisoblash uchun domenga xos dasturlash tili. U C#-ga o'xshash sintaksisga ega va kvant algoritmlarini ishlab chiqish uchun yuqori darajadagi dasturlash modelini taqdim etadi. Q# turli kvant operatsiyalarini qo'llab-quvvatlaydi, masalan, bitta va ko'p qubitli gatellar, o'lchovlar va simulyatsiyalar va u gibridd kvant-klassik algoritmlar uchun klassik hisoblashni qo'llab-quvvatlaydi.

Cirq - bu Python-da kvant sxemalarini yaratish va ishga tushirish uchun open-source kutubxona. U turli xil kvant qurilma platformalarida ishlay oladigan yuqori unumdorlikdagi sxemalarni yaratishga qaratilgan holda moslashuvchan va kengaytiriladigan bo'lishi uchun mo'ljallangan. Cirqni Google Quantum ishlab chiqqan.

PyQuil - bu Rigettining kvant hisoblash platformasidan foydalangan holda kvant dasturlash uchun Python kutubxonasi. U kvant dasturlarini yaratish va ularni real yoki

simulyatsiya qilingan kvant uskunasida ishga tushirish uchun yuqori darajadagi interfeysni taqdim etadi.

IBM Quantum Composer kvant sxemalarini loyihalash uchun drag-and-drop grafik foydalanuvchi interfeysi hisoblanadi. U ta'lim va targ'ibotga e'tibor qaratgan holda, mutaxassis bo'lmaganlar uchun foydalanish uchun qulay bo'lishi uchun yaratilgan.

Ushbu dasturlash tillarining har biri o'zining kuchli va zaif tomonlariga ega va tilni tanlash loyihaning o'ziga xos talablariga bog'liq. Ishlab chiquvchilar kvant hisoblashning hayajonli salohiyatini o'rganish uchun ularning ehtiyojlari va afzalliklariga eng mos keladigan tilni tanlashlari mumkin.

Kvant dasturlashda Python dan Rustni ustunligi

Rust ham, Python ham o'zining kuchli va zaif tomonlariga ega bo'lsa-da, Rust uni kvant hisoblash uchun yaxshi tanlovga aylantiradigan bir nechta xususiyatlarga ega:

Tezlik

Rust - bu tez va samarali bo'lish uchun mo'ljallangan kompilyatsiya qilingan til. Bu uni kvant hisoblash kabi yuqori unumdorlikni talab qiladigan ilovalar uchun juda mos keladi. Bundan farqli o'laroq, Python talqin qilinadigan til bo'lib, u ba'zi ilovalarda sekinroq bo'ladi.

Xavfsizlik

Rust xavfsiz til bo'lib ishlab chiqilgan bo'lib, bufer to'lib ketishi va null ko'rsatgichni yo'qotish kabi keng tarqalgan dasturlash xatolarining oldini olishga yordam beradi. Bu, ayniqsa, hisob-kitoblarning aniqligi muhim bo'lgan kvant hisoblashlarida muhimdir. Python mashhur va moslashuvchan til bo'lsa-da, Rust kabi xavfsizlikka yo'naltirilgan emas.

Parallelizm

Rust kvant hisoblashda muhim bo'lgan parallelizmni qo'llab-quvvatlash uchun mo'ljallangan. Kvant algoritmlari ko'pincha bir vaqtning o'zida bir nechta mustaqil simulyatsiyalarni bajarishni talab qiladi va Rustning parallelizmni qo'llab-quvvatlashi buni samaraliroq qiladi. Python-ning parallelizmni qo'llab-quvvatlashi cheklangan va ko'pincha tashqi kutubxonalaridan foydalanishni talab qiladi.

Low-leveldagi boshqaruv

Rust - bu tizim dasturlash tili bo'lib, u xotirani boshqarish va tizim darajasidagi boshqa operatsiyalar ustidan low-level darajadagi nazoratni ta'minlaydi. Bu maxsus apparat uchun kodni optimallashtirishni osonlashtiradi, bu esa kvant hisoblashda muhim, bunda apparatga xos optimallashtirish ish faoliyatini sezilarli darajada yaxshilashga

olib kelishi mumkin. Python, yuqori darajadagi til bo'lsa-da, low-leveldagi optimallashtirishga unchalik mos kelmaydi.

Qisqa qilib aytganda, Rustning ishlashi, xavfsizligi, parallelligi va low-leveldagi boshqaruvi uni kvant hisoblash ilovalari uchun samarador tanlovga aylantiradi, ayniqsa yuqori unumdorlik va apparatga xos optimallashtirish muhim bo'lgan hollarda.

Kvant hisoblashning salohiyati juda katta, ammo u o'z va'dasini amalga oshirishdan oldin yengib o'tish kerak bo'lgan muhim qiyinchiliklar mavjud. Ba'zi asosiy muammolar qatoriga xatolarni tuzatish zarurati, kvant tizimlarini kengaytirish qiyinligi va texnologiyaning hozirgi holati kiradi. Xatolarni tuzatish juda muhim, chunki kvant tizimlari noise va xatolarga olib kelishi mumkin bo'lgan atrof-muhit omillariga juda sezgir. Kvant tizimlarini kengaytirish qiyinligi qiyin, chunki mazmunli hisoblash uchun zarur bo'lgan qubitlar soni hozirda cheklangan va qubitlar sonini ko'paytirish qo'shimcha noise va xatolarga olib kelishi mumkin.

Ushbu qiyinchiliklarga qaramay, bu sohada sezilarli yutuqlarga erishildi va bir nechta kompaniyalar va tadqiqot guruhlarini kengaytiriladigan kvant hisoblash tizimlarini yaratish ustida ishlamoqda. Ushbu tizimlar keng tarqalib borar ekan, kvant kompyuterlari uchun kod yoza oladigan dasturchilarga talab ortib boradi. Kvant dasturlashni o'rganishga qiziquvchilar uchun Internetda ko'plab resurslar mavjud, jumladan, o'quv qo'llanmalari, onlayn kurslar va amaliyot dasturlari.

Qisqa qilib aytganda, kvant hisoblash shiddat bilan rivojlanayotgan soha bo'lib, kriptografiya, optimallashtirish va simulyatsiya kabi sohalardagi murakkab muammolarni hal qilish uchun ulkan salohiyatga ega. Hali hal qilinishi kerak bo'lgan jiddiy muammolar mavjud bo'lsa-da, so'nggi bir necha yil ichida bu sohada erishilgan yutuqlar ta'sirchan bo'ldi va kvant hisoblashning kelajakdagi istiqbollari hayajonli. Ko'proq tadqiqotchilar va kompaniyalar texnologiyaga sarmoya kiritar ekan, kelgusi yillarda kvant hisoblashlari kompyuter fanining tobora muhim sohasiga aylanishi mumkin.

Kelajakdagi ishlanmalar

Kvant hisoblash sohasi jadal rivojlanmoqda va davom etayotgan tadqiqot va ishlanmalar yanada mustahkam va kengaytiriladigan kvant hisoblash tizimlarini, shuningdek, yanada kuchli va foydalanuvchilarga qulay kvant dasturlash tillarini yaratishga qaratilgan. Kvant dasturlash tillarida davom etayotgan ba'zi ishlanmalarga quyidagilar kiradi:

Yaxshilangan simulyatsiya imkoniyatlari

Kvant hisoblash tizimlarini aniqroq simulyatsiya qila oladigan dasturlash tillarini ishlab chiqish, bu yanada samaraliroq va samarali sinov va optimallashtirish imkonini beradi.

Xulosa

Kvant dasturlash tillari kvant hisoblash kelajagining muhim tarkibiy qismidir. Soha o'sishda va rivojlanishda davom etar ekan, tadqiqotchilar va amaliyotchilar uchun eng so'nggi ishlanmalar va vositalardan xabardor bo'lish juda muhimdir. Kuchliroq va foydalanuvchilarga qulay kvant dasturlash tillarining davom etayotgan rivojlanishi bilan kvant hisoblashning kelajagi istiqbolli ko'rinadi. Kvant hisoblashlari kengroq qo'llanilsa, kvant dasturlash kriptografiya, optimallashtirish va simulyatsiya kabi sohalarga ta'sir ko'rsatadigan kompyuter fanining tobora muhim sohasiga aylanishi mumkin.

ADABIYOTLAR:

1. Потапов А.С. Технологии искусственного интеллекта - СПб: СПбГУ ИТМО, 2010.-218 с.
2. Интеллектуальные информационные системы и технологии: учебное пособие / Ю.Ю. Громов, О.Г. Иванова, В.В. Алексеев и др. – Тамбов: Изд-во ФГБОУ ВПО «ТГТУ», 2013. – 244 с.
3. Ignatyev N.A., Usmanov R.N., Madraximov Sh.F. Berilganlarning intellektual tahlili // O'quv qo'llanma. Toshkent – 2018, 144 b.
4. Асадуллаев Р.Г. Нечеткая логика и нейронные сети: учебное пособие /– Белгород, 2017. – 309 с.
5. Джонс М. Т. Программирование искусственного интеллекта в приложениях – М.: ДМК Пресс, 2011. – 312 с.
6. Mirziyoyev Sh.M. Buyuk kelajagimizni mard va olijanob xalqimiz bilan birga quramiz. – Toshkent: “O'zbekiston”, 2017. – 488 b.
7. Гаврилова Т.А., Хорошевский В.Ф. Базы знаний интеллектуальных систем. /Учебник. СПб.: Питер, 2001. - 384 с.
8. Джоши, Пратик. Искусственный интеллект с примерами на Python.: Пер. с англ. - СПб.: ООО "Диалектика", 2019. - 448 с.
9. <https://www.youtube.com/watch?v=-UlXHPIEVqA>
10. <https://www.youtube.com/watch?v=jHoEjvuPoB8>
11. <https://www.youtube.com/watch?v=X8MZWCGgIb8>
12. <https://www.youtube.com/watch?v=2uV5XwhH6Eg>
13. <https://docs.python.org/3/tutorial/>
14. <https://www.tutorialspoint.com/python/index.htm>
15. <https://www.w3schools.com/PYTHON/>
16. <https://www.learnpython.org/>

17. <https://jobtensor.com/Tutorial/Python/en/Introduction>
18. <https://www.programiz.com/python-programming>
19. <https://realpython.com/>
20. <https://www.javatpoint.com/python-tutorial>
21. <https://pythonworld.ru/kursy/free.html>
22. <https://stepik.org/course/67/promo>
23. <https://netology.ru/programs/python-free>
24. <https://itproger.com/course/python>
25. <https://gb.ru/courses/112>
26. <https://www.udemy.com/ru/topic/python/free/>
27. <https://qna.habr.com/q/461609>