Документация на проект:

Simulation Q

Регистрационен № …..

1. Тема

На прага сме на нова ера в компютърните възможности, нова ера, в която ще трябва да преосмислим своето разбиране не само за изчислимост, но и сигурност, телекомуникация, анализ на данни и много още. Квантовата ера. Следващата наша „малка стъпка за човек“ ще бъде работата на такъв компютър на стайна температура,но „големия скок за човечеството“ ще бъде практическото имплементиране на тези машини в лаборатории, или дори внедрени в домовете на хората.

Фундаментално квантовите компютри работят различно от класическите компютри, затова трябва да подходим по различен начин. Въпреки че тези машини са все още далеч от това да се намират във всеки дом, ние трябва да сме подготвени за този момент. Когато това се случи, трябва вече да сме разработили работещи алгоритми. Точно в тази стъпка ще ни бъдат най- полезни симулациите като „Simulation Q“. Те са начин, от една страна, да разработим програми без да имаме достъп до физическата машина, и от друга страна, да измерваме производителността на същите тези машини, за да можем точно да определим момента, в който квантовите ще бъдат по- ефективни от класическите компютри.

1. Автори

Имена: Цветелин Костадинов Цецков,

ЕГН: 0143123249

Адрес: ул. Мария Димова №31, гр. Монтана

Телефон: 00 359 888 16 27 58

E-mail: [tsvetelinkostadinovts@gmail.com](mailto:tsvetelinkostadinovts@gmail.com)

Училище: ПМПГ „Св. Климент Охридски“ – Монтана

Клас : XII

1. Ръководител

Имена: Илиян Георгиев Ценков,

Телефон: 00 359 885 92 79 55

E-mail: [i\_cenkov@yahoo.com](mailto:i_cenkov@yahoo.com)

Длъжност: Старши учител по информатика и информационни технологии

1. Резюмe
   1. Цели

Разработката Simulation Q има за цел да предостави възможност за разработване на алгоритми предназначени за имплементиране в квантов компютър. Тези машини са изключително нестабилни, което налага само тяхното експериментално, а от скоро и много тясно практическо, изучаване. Simulation Q позволява на всеки разработчик да борави теоретично с такъв компютър, без да се налага да има физически достъп до такъв. Това би намалило разходите за разработка на квантови програми в бъдеще, когато квантовите машини стигнат до пазара.

Със симулация също можем да следим колко се различават квантовия компютър и класическия по отношение на производителността и изчислителните им способности. Така бихме могли да определим точно в кой момент от развитието си квантовата машина надминава класическата и с каквъ коефициент по отношение на времето.

Докато тези машини са нерентабилни за широката аудитория, единствения начин за развитие на програми са симулациите като Simulation Q, също така остават и ключалката, през която можем да надникнем в бъдещето и да се опитаме да се подготвим за него. Внедрявайки Simulation Q, или поне теорията за квантовите компютри, в образователната ни система ще подготвяме децата за бъдещето, което не изглежда толкова далеч с последните разработки в областта.

* 1. Процес на работа

1. Идеен етап

Оформи се идеята, че има нужда от подобна българска разработка, която да е широко достъпна. Реши се един основен въпрос- категорията на проекта да бъде приложна програма или интернет приложение. Класифицира се като приложна програма, но се изгражда с възможност за интегриране в интернет сайт като backend.

1. Полагане на математически основи

Квантовите проблеми изискват квантово обяснение. Като материали бяха използвани много статии и разработки по темата, някои полезни, други не, беше установено, че по естество квантовите явления са далечни от всяка аналогия с познатия ни свят, те могат да бъдат полезни като въведение, но на нивото на Simulation Q единственият език, на който може напълно да се предаде значението на дадена функционалност е езикът на математиката.

В този етап бе разработен математическият модул на симулацията, който представлява линейна алгебра- операции върху матрици и вектори в комплексната равнина. Това представляваше и един от първите етапи на затруднение поради факта, че трябваше да се издирят подходящи учебници и да се изчете литературата по въпроса.

1. Моделиране на квантовите единици за изчисление

Kвантова единица за съхранение на данни и изчисление(наричана по-долу qubit- quantum bit или кюбит на български). Битовете в класическите компютри могат да бъдат в 2 състояния- 1 или 0, докато кюбитовете могат да бъдат в смесено състояние, в което се крие тяхната изчислителна мощ

Множество статии представят кюбита като състояние на всички възможни компинации, но това е грешна аналогия. В квантовия компютър те са в смесено състояние, имат 2 комплексни числа асоциирани със състоянието си, които 2 числа ние манипулираме, за да получим резултатът си. Един кюбит преставен чрез анотацията на Пол Дирак като:

**ψ** – разглежданият кюбит

**α** – комплексната вероятност кюбитът да произведе стойност |0>

**β** – комплексната вероятност кюбитът да произведе стойност |1>

Допълнително съкращенията(кетове) |0> и |1> означават съответно 2та базови вектора - и

Вероятността се определя като се повдигнат на квадрат коефициента пред съответния кет - за |0> и за |1>

Като последното условие, за да съществува този кюбит е:

Това условие гарантира, че общата вероятност за произвеждане на |0> или |1> е точно 100%

За точно това явление не можем да намерим аналог в ежедневния ни живот, защото то се проявява само на микроскопични нива, които ние можем само да анализираме, без да виждаме.

Компютърния модел изискваше прецизно планиране и стриктно използване на функционалната парадигма, защото на този етап трябваше да се изгради много базова функционалност и проверки за грешни данни, за да се избегнат грешки в бъдеще. Веднага щом се завърши модела на кюбита бе разработен и базов алгоритъм за неговия разпад, за да можем да анализираме данните, които получаваме.

Приемайки, за по-лесна демострация, че α и β са реални чесла, можем да представим кюбита като вектор ориентиран в единична окръжност, или както се нарича в областта на квантовите изчисления – Блох окръжност(кръстена на Феликс Блох, швейцарски физик).

Елемент 1

Елемент 2

Елемент 3

Елемент 4

Фиг. 1

На фиг. 1 можем да разгледаме различни репрезентации на няколко кюбита

* Елемент 1 – кюбит с уравнение Което значи, че има 100% вероятност да произведе резултат |1>
* Елемент 2 – аналгично с Елемент 1, но с уравнение
* Елемент 3 - В този случай можем да разгледаме примера с и . Тогава можем да определим вероятността кюбитът да произведе |0> като  
   и , от тук следва, че кюбитът има 50% вероятност да произведе |0> и 50% да произведе |1>
* Елемент 4- общия случай на кюбит в смесено положение

Ако разширим дефиниционната област на , Блох окръжността става Блох сфера и имаме още една посока за работа, по която можем да правим трансформации

1. Имплементиране на базови операции

Базовите операции в квантовия компютър(от англ. Quantum gates) математически са представени чрез матрици. Прилагането на операциите се извършва чрез просто умножение на транспонирания вектор на кюбита с матрицата на оператора.

Операциите, обаче, също имат специфично ограничение- а именно да са обратими(не матриците, а операциите). Това означава, една и съща операция приложена двукратно да връща кюбита в начално състояние:

H

H

На фигура ....... е показано прилагането на операцията Hadamard два пъти връщайки кюбита в първоначалното му състояние. Чрез уравнение можем да го представим така:

Ако имаме операция с изражение равно на матрицата **.** Тогава:

, където е единичната матрица

https://upload.wikimedia.org/wikipedia/commons/thumb/1/1a/Hadamard_gate.svg/150px-Hadamard_gate.svg.pngПримери:

Предишно споментатата операция Hadamard- кръстена на Jacques Salomon Hadamard, виден математик с еврейски произход и френско гражданство. Тя е най- лесния начин да се вкара един кюбит в максимално смесено състояние(както на един, така и на няколко кюбита) – Бел състояние. Неговата операция е с уравнение:

Или

Още едни основни операции са Pauli-X/Y/Z

https://upload.wikimedia.org/wikipedia/commons/thumb/2/23/Qcircuit_NOT.svg/150px-Qcircuit_NOT.svg.png

Отрицание - |0> става |1> и |1> става |0>

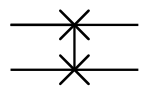
Интересното при тях е, че:

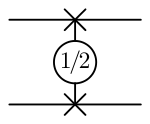
Има и някои изключения от правилото за обратимост, не точно, но при тях е необходимо да се приложат повече пъти за да достигнем до началното състояние. Такава операция е

https://upload.wikimedia.org/wikipedia/commons/thumb/b/b1/Qcircuit_SqrtNot.svg/150px-Qcircuit_SqrtNot.svg.pngТя е корен квадратен от Pauli-X, което й дава уравнението

Могат да се конструират и други радикали от операциите, ако те удовлетворяват условията на квантовите операции(да са обратими след определен брой апликации).

Освен операции върху един кюбит съществуват и операции върху няколко такива, като за по- кратко и по аналогия с класическите компютри множество кюбити се наричат регистър. Нужно е отново да се отбележи, че операциите също се представят чрез матрици и трябва да отговарят на същия критерии както и критерия за обратимост.

SWAP- операцията е точно такава. Тя оперира върху 2 кюбита, като практически разменя 2 стойности. Тя има матрица:



Тя също има , но обяснението зад тази операция е по- сложно, тя също разменя стойностите, но наполовина, т. е. вкарва 2 кюбита в смесено състояние, но също така и преплита техните състояния( преплитането на състояния ще бъде обяснено по- долу)

Операциите върху няколко кюбита имат свое собствено подсемейство – контролираните операции, т.е. един(или няколко кюбита) служат като контрол върху това дали дадена операция ще бъде приложена на друг кюбит.

, където CO(регистър) е контролирана операция (Controlled Operation) и О(кюбит) е просто операцията.

Концептуално CNOT- Controlled NOT оперира върху 2 кюбита. Когато първият(контролиращият) е |1> тогава вторият(върху който се оперира) ще бъде обърнат(или с други думи - да му приложен Pauli-X, затова другото име на операцията е cX ). Матрицата е:

Има интересна закономерност, която може да бъде приложена за разработване на нови операции, а тя е:

Ако искаме да направим една унарна операция контролирана, тогава можем да приемем, че унарната операция има матрица , тогава можем да вземем функцията C(U) с матрица

На този етап бяха поставени основите на възможността да се изнасят подпрограми(еквивалентът на методите в съвременното програмиране), но не бяха имплементирани напълно с оглед оптимизиране след имплементация на преплитането на състоянията.

1. Преплитане на състоянията на 2 кюбита

Някои операции върху регистър от кюбитове, освен че ги трансформират в смесено състояние, ги преплитат, т.е. стойността на единия кюбит е директно зависим от стойността на другия, независимо от време и пространство. Този феномен е изучаван от Албърт Айнщайн, като от него е наречен „Spooky action happening at a distance“(странно действие от разстояние). Причината за това е било схващането, че информация не може да бъде пренесено по- бързо от скоростта на светлината. В днешно време вече знаем, че при този процес не се пренася информация, но все още не е добре изучен.

Преплитането на състояния може да се случи само при регистър от 2 или повече кюбита, като обикновено те се предтавят отново в кетове. Регистър от 2 кюбита може да бъде представен така:

(Точно тук е много явно превъзходството на кюбитовете пред класическите битове- с 2 кюбита можем да представим 4 състояния, и това число расте експоненциално – с n кюбита можем да представим състояния)

Представени чрез уравнение тези регистри бързо стават много дълги, затова често се използва тензорното произведение на векторите

Например за 2 кюбита:

Тогава:

Това е удобен и интуитивен начин да се представи информацията съхранена в тези 2 кюбита.

Но вектор изглеждащ така:

2та кюбита са преплетени, ако разгледаме уравнениетоим:

Вижда се, че стойността на единия кюбит е директно зависима от стойността на другия( в случая и двата кюбита ще имат една и съща стойност )

Това квантово явление е трудно за моделиране в съвременните компютри, въпреки тяхната изчислителна мощ, не можем да имитираме непредвидимостта на природата. Затова алгоритъмът за разпад на регистър търпи постоянно усъвършенстване и оптимизиране.

Често тези коефициенти(пред кетовете) се наричат апмлитуди. Поради естеството на микрочастиците, те могат да се разглеждат като вълни и частици. Въвеждането на кюбитове в регистри ни позволява да манипулираме амплитудите им така, че да интерферират деструктивно в ненужните стойност и конструктивно в полезните стойност. Това пренася намирането на правилния резултат от вероятностен процес в детерминистичен.

* 1. Архитектура

Simulation Q е изграден от 2 модула:

* Математически
* Квантов(симулационен)

В математическия модул са имплементирани всички структури от линейната алгебра- матрици, вектори, както и по- прости структури като комплексни числа. Всички обекти обвързани с техните операции. Този модул следва функционалната парадигма като се фокусира предимно върху методите за трансформация, а не върху самите структури. В дългосрочен план това би позволило лесното усъвършенстване на всички свързани с математическия модул. Алгоритмите за операциите върху структурите са оптимизирани, за да се сведе до минимум времето необходимо за изчисления( напр. умножението на 2 матрици, с други думи прилагането на операция върху кюбит ).

Другият модул е квантовият, който моделира квантовите частици, регистрите както и операциите върху тях. Работата по този модул следва ООП парадигмата, тъй като е по- близа до потребителя и би улеснило бъдещото развитие. Този модул лежи изцяло върху вече изграденият математически модул. Най- трудният алгоритъм за имплементиране в тази част е разпадът на регистър след като състоянието на няколко кюбита е било преплетено.

Потребителски интерфейс

Математически модул

Квантов модул

* 1. Реализация

За реализацията на този проект бе избран езикът за програмиране Java, тъй като е широко разпространен. Възможността симулацията да може да бъде пусната на всяка машина е от критично значение, за което Java допринася със своя байткод. За компилатор бе избран OpenJDK със своя отворен лиценз и бързи темпове на развитие. Предоставени са възможности да се изпробват експериментални или дори нестабилни функционалности, които биха увеличили продуктивността. Друга причина за избора на Java е строгата типизаця, която увеличава времето за компилация, тъй като се правят множество проверки, но това става за сметка на бързина на по- следващ етап, а по време на изпълнение симулацията се нуждае от всеки последен ресурс, за да се изпълни по- бързо, поради големия брой изчисления, които трябва да се направят.

* 1. Описание на приложението

Минимални изисквания към машината:

* Java
* 2GB RAM

След двойно щракване върху иконката на симулацията ще се стартира и след необходимото зареждане ще бъде готова за работа.

Приложението е оформено като прозорец с опции за drag-and-drop функционалност.

Начален екран

Резултати

Стойности

Ако потребителят желае може да настои колко рапада да има, като зависимостта на точността към броя на разпадите е правопропорционална.

Друга опция е потрбителят да експртира файл с всеки един разпад, в този файл се записва номерът на разпадът, както и неговият резултат. Много полезна функция, с която след правене на определена ситуация могат да се анализират данните със скриптов език като например Python. Тази функция би била също полезна, ако машината, на която се пуска симулацията не е с добри параметри, тогава този файл може да се използва като запис в база данни, и след няколко разпадни цикъла се анализира резулатът.

Всяка операция има подсказка разказваща малко за функционалността на съответната операция, нейната матрица и малко за човекът на който е кръстена(ако има такъв).

1. Заключение

Квантовите симулации тепърва ще навлизат в ежедневието на всеки един софтуерен разработчик. Физическите машини са изключително скъпи и към този момент нестабилни. Симулациите са рентабилен заместител и двигател на бъдещи разработки. Към моментът на започване на проектът подобни разработки нямаше, или бяха също толокова нестабилни както и физическата алтернатива. В последващите месеци IBM и Google обявиха, че започват проучвания в областта на квантовата информатика, а по- късно пуснаха и свои симулации. Това, което отличава   
Simulation Q от останалите разработки е, че не се нуждае от връзка с интернет   
(в случаите на IBM и Google) или директна връзка с физическата машина. Simulation Q има възможността да бъде пусната на суперкомпютър или дори паралелизирана между няколко такива, за да се направи своеобразно състезание кай компютър би решил даден проблем по- бързо.

От към пътища за развитие симулацията би могла да тръгне по няколко пътя, всеки от които със своите плюсове и минуси. Симулацията има още малко място за оптимизация, която задължително ще бъде правена след имплементацията на всяка нова функционалност до постигане на оптимално време за изпълнение.

Друга възможност, която би отвела разработването на квантови алгоритми до ново ниво е въвеждането на скриптов език в симулацията, което ще приближи симулацията до теоретичния процес на работа на един квантов разработчик.

По своята същност симулацията е образователна разработка с цел „да разработваме алгоритмите за утре, днес“. Точно с това мото авторът добавя нови функционалности, които да са в услуга на бъдещите разработчици.