

HOE PROGRAMMEER JE EEN KWANTUM-COMPUTER?

Hoewel kwantumcomputers op een manier werken die heel anders is dan bij reguliere computers, is het niet zo moeilijk om ze te programmeren. In dit artikel vatten we de huidige stand van de technologie in kwantumprogrammering samen.

door Maris Ozols en Michael Walter beeld Shutterstock

WE WETEN HOE WE MILJOENEN TRANSISTORS OP ÉÉN ENKELE CHIP MOETEN ZETTEN EN WE KUNNEN MILJOENEN REGELS CODE SCHRIJVEN DIE SCHIJNBAAR VLEKKELOOS OP DEZE CHIP DRAAIEN. Is er nog iets echt nieuws te ontdekken over het maken van berekeningen?

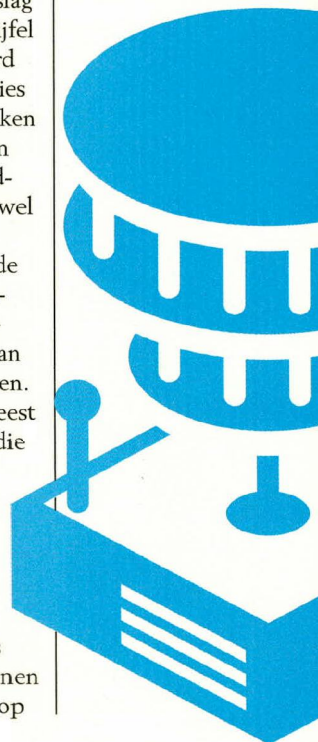
De moderne berekeningsmethode is in de jaren dertig opgesteld door de Engelse wiskundige Alan Turing. Ondanks het enorme succes heeft de theorie van Turing een cruciaal ingrediënt weggelaten: de natuur. Inderdaad, computers zijn geen abstracte machines, maar hebben een fysieke belichaming. Aan de ene kant is dit een beperking vanwege de fysieke grenzen die meespelen. Het kan echter ook een voordeel zijn, omdat de ontdekking van nieuwe theorieën over fysica tot nieuwe manieren van berekening kan leiden.

KWANTUMREVOLUTIE

De eersten die vermoedden dat het maken van digitale berekeningen draaiend

op transistoren misschien niet het laatste antwoord is, waren natuurkundigen. Ze begonnen de aannames die ten grondslag lagen aan de theorie van Turing in twijfel te trekken nadat ze hadden geprobeerd om met atomen vergelijkbare simulaties uit te voeren. Deze vergelijkingen maken deel uit van een theorie die in de jaren twintig was ontwikkeld en die bekendstaat als de kwantummechanica. Hoewel veel briljante geesten, waaronder Einstein, twijfels hebben gehad over de ware betekenis ervan, geeft kwantummechanica de meest nauwkeurige beschrijving van de innerlijke werking van het universum op de schaal van atomen. Het heeft daarom de status van de meest grondig geverifieerde fysieke theorie die de mensheid ooit had.

De kracht van kwantummechanica blijft echter nog steeds grotendeels ver van ons verwijderd, omdat we niet weten hoe we deze op een computer moeten simuleren. Kan zo'n theorie ergens goed voor zijn als we haar voorspellingen niet eens kunnen berekenen? Een ingenieus antwoord op



HOE ZIET EEN KWANTUMPROGRAMMA ERUIT?

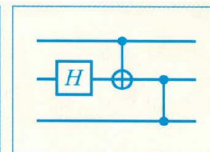
Een veelbelovende toepassing voor toekomstige quantumcomputers is het simuleren van de dynamiek van complexe fysieke systemen, zoals chemische moleculen en kwantummaterialen. Al vandaag de dag bestaan er bibliotheken (zoals de open-sourcebibliotheek OpenFermion) die kwantumalgoritmen voor deze problemen implementeren. Programmeurs kunnen dergelijke bibliotheken gebruiken door een eenvoudige reeks opdrachten op hoog niveau aan te roepen:

```
...
molecule = load_molecular_data('H2')
E = estimate_ground_energy(molecule)
...
```

Deze commando's op een hoger niveau worden door quantumcompilers naar primitieven op een lager niveau vertaald. Uiteindelijk komt het uitvoeren van het programma neer op het toepassen van een reeks elemen-

taire kwantumpoorten die direct op quantumhardware kunnen worden uitgevoerd. Hier is een codefragment van de kwantumassemblage, samen met een diagram van de bijbehorende kwantumschakeling:

```
...
H q[2]
CNOT q[1], q[2]
CZ q[2], q[3]
...
```



deze vraag werd gegeven door Richard Feynman en anderen die in de jaren tachtig suggereerden dat we de theorie moesten inzetten voor het bouwen van betere computers. Onderzoeken hoe de aard van kwantummechanica tot betere berekeningsmogelijkheden kan leiden, is nu een enorm wetenschappelijk onderzoeksgebied geworden en ook grote techbedrijven zijn er steeds vaker mee bezig. Er zijn al kleine prototypen van quantumcomputers gebouwd en de eerste generatie quantumprogrammeurs racet om hun moedertaal als de dominante taal te vestigen op het nieuwe en nog steeds grotendeels onbewoonde quantumcontinent.

KWANTUM-PROGRAMMEREN

Kwantumcomputers slaan een reeks kwantumbits of qubits op. Net zoals gewone computers eenvoudige bewerkingen kunnen uitvoeren, zoals AND en NOT door het gebruik van logische poorten, kan een quantumcomputer worden geprogrammeerd door kwantumpoorten te gebruiken. Elke poort voert een elementaire bewerking uit op één of twee qubits tegelijk. Hoewel logische poorten kunnen worden gespecificeerd in termen van hun waarheidstabel (bijv. 1 AND 0 = 0, enz.), worden kwantumpoorten beschreven door kleine meetgegevens: een 2 x 2-matrix voor een enkele qubit en een 4 x 4-matrix

voor twee qubits. Door een geschikte reeks kwantumpoorten toe te passen, kan men een willekeurig gecompliceerde kwantumschakeling samenstellen. Bij het ontwikkelen van geavanceerdere software wordt het snel moeilijk om de implementatie ervan als een reeks elementaire poorten te specificeren.

In plaats daarvan maken programmeurs quantumprogramma's waarbij ze geavanceerdere bouwstenen inzetten, zoals de zoekfunctie van Grover of de quantum Fourier transform, of zelfs functionaliteit van een hoger niveau als subroutines voor kwantumoptimalisatie.

Kwantumprogramma's worden vaak gespecificeerd in een taal op een hoog niveau, zoals Python, en er zijn verschillende concurrerende platforms die over de hele wereld worden ontwikkeld, zoals Google Cirq, IBM Qiskit, Microsoft Quantum Development Kit, ProjectQ en Rigetti Forest.

Wat ze gemeen hebben, is dat op de een of andere manier de high-level aanwijzing wordt vertaald in instructies op een lager niveau, vergelijkbaar met de assembleertaal, die direct een quantumcircuit vertegenwoordigen (IBM's OpenQASM en Rigetti's Quil zijn twee voorbeelden). Dit compilatieproces bevat meestal verschillende fasen van optimalisatie die rekening kunnen houden met de doelarchitectuur. Zo ondersteunt bijvoorbeeld elk type quantumhardware gewoonlijk slechts een klein aantal kwantumpoorten, van waaruit



WORDEN KWANTUM- COMPUTERS OOI EEN RESTAURANT MET EEN MICHELINSTER?

alle andere bewerkingen moeten worden gesynthetiseerd. Deze poorten zijn mogelijk niet altijd direct toepasbaar op willekeurige doel-qubitparen vanwege connectiviteitsbeperkingen, waardoor een extra stap nodig is om het circuit aan te passen aan de gegeven topologie. Om rekening te houden met verschillende onvolkomenheden in de hardware, zullen toekomstige kwantumcompilers ook een extra stap van foutcorrectie implementeren.

Naast het uitvoeren van kwantumprogramma's op echte kwantumhardware, is het vaak ook handig om kleine tests uit te voeren op een simulator. In een echte kwantumcomputer zijn, vanwege eigenaardigheden van de kwantummechanica, een zeer groot aantal iteraties vereist om de toestand van zijn kwantumbits te bepalen in elk stadium van de berekening. Met simulators kan de programmeur de status van de kwantumbits direct inspecteren, waardoor foutopsporing en introspectie van kwantumprogramma's wordt vergemakkelijkt. Dergelijke simulaties werken natuurlijk alleen voor maximaal ongeveer 50 qubits, zelfs bij het gebruik van moderne supercomputers, en ze kunnen de computationele snelheid niet ontgrendelen die alleen toegankelijk is voor echte kwantumhardware.

TOEKOMST VAN KWANTUMCOMPUTERS

Om een idee te krijgen over de toekomst van kwantumcomputers, is het nuttig om te kijken naar modern high-performance computing. Deze laatste maakt gebruik van gespecialiseerde hardware en software en wordt ingezet voor specifieke problemen. Het is inderdaad zo dat de beschikbaarheid van enorme parallele rekenkracht in de vorm van grafische kaarten (GPU's), gekoppeld aan speciale software (zoals

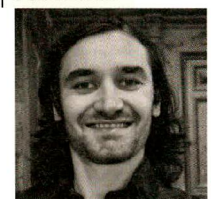
Google's TensorFlow), buitengewoon succesvol is geweest in het versnellen van machinelearningberekeningen. Dat heeft de aanhoudende revolutie in machine learning mogelijk gemaakt. Het is goed voor te stellen dat kwantumcomputers een vergelijkbare rol zullen spelen, met een enorme rekenkracht voor gespecialiseerde toepassingen. Maar naast technologische uitdagingen, zoals decoherentie en ruis, is misschien de moeilijkste uitdaging voor het programmeren van toekomstige kwantumcomputers in de eerste plaats niet welke programmeertaal te gebruiken, maar welk programma te schrijven!

Het is inderdaad een fundamentele uitdaging om nieuwe en geavanceerde algoritmen te vinden die gebruik kunnen maken van het verrassende computationele voordeel dat kwantummechanica biedt. Het Nederlandse onderzoekscentrum QuSoft, een joint venture van de Universiteit van Amsterdam, CWI en de Vrije Universiteit Amsterdam, is een wereldleider in de ontwikkeling van nieuwe kwantumsoftware en -algoritmen. QuSoft-onderzoekers van Science Park, Amsterdam, onderzoeken al meer dan 20 jaar de mogelijkheden en beperkingen van kwantumcomputers en ontdekken nieuwe algoritmen voor optimalisatie, machine learning, communicatie en cryptografie. Als we het veld van kwantumcomputers zouden vergelijken met een nieuw restaurant dat over enkele jaren wordt geopend, kunnen we de potten en pannen al in de keuken horen klinken en zien we een aantal drukke chef-koks strijden om de leidende rol. Het voorgerechtge-deelte op het menu is grotendeels over-eengekomen, maar de juiste technologie en ingrediënten voor het hoofdgerecht zijn nog steeds in de maak – het vereist veel vindingrijkheid en experimenten om ze samen te voegen!

Zullen kwantumcomputers ooit een restaurant met een Michelinster worden? We weten het niet. Maar we kunnen zeker voelen dat de geur uit de keuken komt – het ruikt heerlijk! 🍴

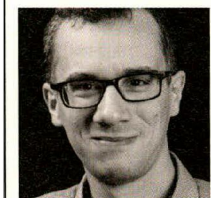


AUTEUR



MARIS OZOLS

Universitair docent aan de Universiteit van Amsterdam en senior researcher bij QuSoft. Maris promoveerde aan de University of Waterloo (Canada) en was postdoctoraal onderzoeker bij IBM en de University of Cambridge. Zijn onderzoek richt zich op kwantumalgoritmen en kwantuminformatietheorie.



MICHAEL WALTER

Universitair docent aan de Universiteit van Amsterdam en senior researcher bij QuSoft. Michael promoveerde aan de ETH Zürich en was postdoctoraal onderzoeker aan de Stanford University. Zijn onderzoek richt zich op kwantuminformatietheorie en de verbanden ervan met wiskunde, fundamentele natuurkunde en informatica.

