Report IBM

Brokerage session

Davide Camino

November 3, 2025

1 Post quantum encryption

Io di crittografia, e sicurezza in generale, ne so molto poco. Forse proprio per questa mia carenza ho trovato particolarmente interessante il discorso che comunque non si è mai addentrato in tecnicismi particolari.

Nonostante non sia sceso nei dettagli Nicola Bertoli ha chiarito alcuni punti che non aveva potuto approfondire la mattina, ma soprattutto ci ha mostrato alcune risorse per approfondire lo studio della crittografia post quantistica. Queste ultime secondo me sono la cosa più rilevante del suo intervento.

1.1 Risorse teoriche

Abbiamo parlato molto brevemente della matematica che sta alla base dei problemi sui reticoli, mostrando di sfuggita alcune definizioni degli spazi presi in considerazione e la definizione formale dei problemi di minimizzazione.

Oltre al materiale preparato per la presentazione della mattina abbiamo usato anche queste slides che non sono di IBM, ma dell'università di Waterloo. Essendo slides di un corso universitario scendono molto più nel dettaglio, soprattutto dal punto di vista matematico, definendo formalmente tutti i problemi che abbiamo visto la mattina e altri che sono sempre considerati quantum-safe.

Sono parecchio impegnative, ma non mi è parso che richiedano delle competenze particolari di crittografia per essere lette. Ogni problema presentato è definito e spiegato in maniera, seppur concisa, chiara.

Come dicevo il discorso di Nicola è stato comunque molto introduttivo e non abbiamo visto nessuna definizione o dimostrazione, però partendo dalle slides dell'università di Waterloo penso ci siano tutti gli spunti per approfondire a piacimento.

1.2 Risorse pratiche

L'intervento di Nicola è stato divertente anche perché abbiamo giocato a fare gli "hacker". Abbiamo utilizzato gli strumenti sviluppati da Open Quantum Safe

un progetto open-source che fa parte della Linux Foundation e Wireshark per monitorare i pacchetti scambiati sulla rete.

A Open Quantum Safe hanno implementato gli algoritmi di crittografia post-quantum in una libreria che hanno chiamato liboqs. Hanno poi integrato questi algoritmi in una serie di applicazioni sia lato server che lato client; qui la lista.

Open Quantum Safe mette anche a disposizione una serie di server ngnix (a questo indirizzo) a cui fare richieste con i nuovi protocolli. Abbiamo usato la versione di `curl` post-quantum per fare richieste a uno di questi osservando con Wireshark cosa esattamente stesse viaggiando sulla rete.

Infine sempre con i tool di Open Quantum Safe abbiamo cifrato e decifrato messaggi per valutare le performance di algoritmi classici e post-quantum. È risultato che gli algoritmi post-quantum hanno performance confrontabili con gli algoritmi classici, nonostante la chiave per cifrare sia notevolmente più lunga.

Nulla di nuovo rispetto a ciò che è stato detto la mattina, ma sapere che esiste una implementazione liberamente consultabile e vedere gli algoritmi che girano è sempre bello.

1.3 Fully Homomorphic Encryption

Alla fine ci ha mostrato, più per curiosità che per altro, questo testo che parla di crittografia omomorfica: sostanzialmente si è scoperto che certe operazioni sia booleane che aritmetiche si possono fare anche sui dati cifrati.

Nel caso studio che ha presentato l'idea era di cifrare delle immagini sensibili, ad esempio lastre ai polmoni, mandarle cifrate a una rete neurale, ottenere una risposta cifrata, decifrarla in locale per stabilire se il paziente avesse o meno il COVID.

2 Quantum Computing

Luca Crippa non si era preparato nulla da presentare, quindi ha parlato molto meno e il discorso è stato più che altro guidato dalle nostre domande.

2.1 Come funziona il ciclo di rilascio di QISKIT?

Questa e la successiva sono domande di Mario, immagino mosse dalla fatica nel configurare l'ambiente python.

QISKIT, da quanto ho capito, funziona un po' come android, c'è una base open-source a cui collaborano IBM e la community. Sopra questa c'è una parte proprietaria di IBM. Per dettagli guardare la domanda sotto.

Le due componenti hanno cicli di rilascio indipendenti, ma si garantisce che il core di QISKIT sia compatibile con gli addon premium di IBM.

Anche la parte gratuita (che sia open o proprietaria) è divisa in componenti e ogni modulo riceve gli aggiornamenti quando sono pronti, senza curarsi troppo di rompere delle dipendenze. Per questo adesso il simulatore non funziona. Questo disallineamento dei pacchetti è dovuto principalmente a uno sviluppo

molto rapido della componente core, ma tutti i componenti dovrebbero essere armonizzati nelle versioni successive.

QISKIT adesso è alla versione 2.x, ma dalla versione 1.0 mantiene la retro-compatibilità, anche se IBM spinge per astrarre sempre di più dalla macchina quantistica per avere sempre più funzioni pre-pronte da mettere direttamente dentro python. Il flusso dei dati dovrebbe diventare più o meno così:

1. elaboro classicamente fino a ottenere un problema di minimizzazione;
2. invoco la funzione di libreria `quantum_optimizer(myProblem)`;
3. termino l'elaborazione su hw classico.

2.2 Quanto è open QISKIT?

Qui il discorso diventa fumoso, ma guadando anche la documentazione dovrei avere fatto il punto. Segue quindi l'elenco delle componenti principali di QISKIT con relativa descrizione

- *SDK Qiskit* completamente open (contiene anche il transpiler)
- *Qiskit Runtime* sono open il lato client e alcuni software per la mitigazione degli errori
- *Qiskit Serverless* proprietario
- *Qiskit Function* proprietarie e accessibili solo da clienti Premium, Flex e On-Prem

Di tutto questo siamo interessati principalmente alle *Qiskit Function*, perché generano dei circuiti evidentemente molto ben realizzati. Le *Qiskit Function* si dividono in:

Funzioni di circuito: servizi che comprendono tecniche di transpilazione, soppressione degli errori, attenuazione degli errori e post-elaborazione che prendono in input circuiti astratti e osservabili di misura desiderati

Funzioni applicative: servizi che comprendono interi flussi di lavoro quantistici, dalla mappatura del classico al quantistico, all'ottimizzazione per l'hardware, all'esecuzione sull'hardware e alla post-elaborazione.

Ad esempio il flusso di lavoro descritto prima potrebbe sfruttare una funzione applicativa.

Mario ha allora chiesto se si potesse fare una sorta di reverse engineering di queste funzioni, in particolare se si potesse osservare il circuito generato prima di eseguirlo. Luca non era sicuro della risposta e io non ho trovato molto sulla pagina da cui ho preso le altre informazioni.

Il riassunto è che le funzioni applicative mascherano completamente tutta l'elaborazione; di queste molto probabilmente non c'è modo di vedere il circuito.

Delle funzioni circuito in teoria dovrebbe essere possibile, queste infatti non mascherano completamente l'elaborazione, ma siccome generano circuiti già transpilati il risultato potrebbe essere talmente complesso da risultare poco utile. Inoltre per ottenere il circuito teorico (non adattato alla specifica QPU) si dovrebbe effettuare la transpilazione inversa.

2.3 Come si misurano effettivamente le performance?

All'interno di IBM sono state proposte diverse metriche per la valutazione delle performance. Queste ultime non devono tenere solo in considerazione il numero di *q-bit*, ma anche l'affidabilità di questi e il tempo necessario a implementare il circuito quantistico.

Attualmente IBM usa i CLOPS (Circuit Layer Operations Per Second) una misura correlata a quanto velocemente la QPU riesce a eseguire un circuito. In particolare i CLOPS saranno tanto più alti quanti più gate si riescono a implementare al secondo. I gate si implementano attraverso treni di impulsi nelle microonde, quindi più treni di impulsi si riescono a generare e spedire sui *q-bit* designati al secondo, maggiore sarà il numero di CLOPS.

2.4 Esistono metriche standard per le performance?

No.

Sostanzialmente ognuno usa la metrica più vantaggiosa. Prima dei CLOPS, IBM usava il Quantum Volume, una misura correlata al numero di *q-bit* e alla loro precisione.

Dato che la tecnologia di IBM, però, punta tutto sulla velocità di esecuzione, una metrica come il Quantum Volume li svantaggiava rispetto a tecnologie con meno *q-bit* e più lente ma molto più precise.

Probabilmente fino a quando un ente esterno non imporrà una standardizzazione, ogni azienda utilizzerà la metrica più conveniente alla particolare tecnologia che sviluppa.

3 EXTRA: il mio Quantum-Computer

Non serve essere milionari per avere in casa un quantum computer. Appena prima di lasciarci ci hanno fatto vedere questo sito dove si possono scaricare le istruzioni e ordinare direttamente i pezzi lego per costruire il proprio modello dei computer di IBM. Alcuni dei modelli sono stati pensati proprio da Luca (P.S. i modelli grandi sono sul migliaio di pezzi, quindi sono solo parzialmente più economici di quelli veri).

Per una versione funzionante invece ci hanno fatto vedere questo sito che contiene le istruzioni per la stampa 3D del modello, con lo spazio per un raspberry-pi per eseguire una versione custom di Raspberry Pi OS che emula il quantum computer.