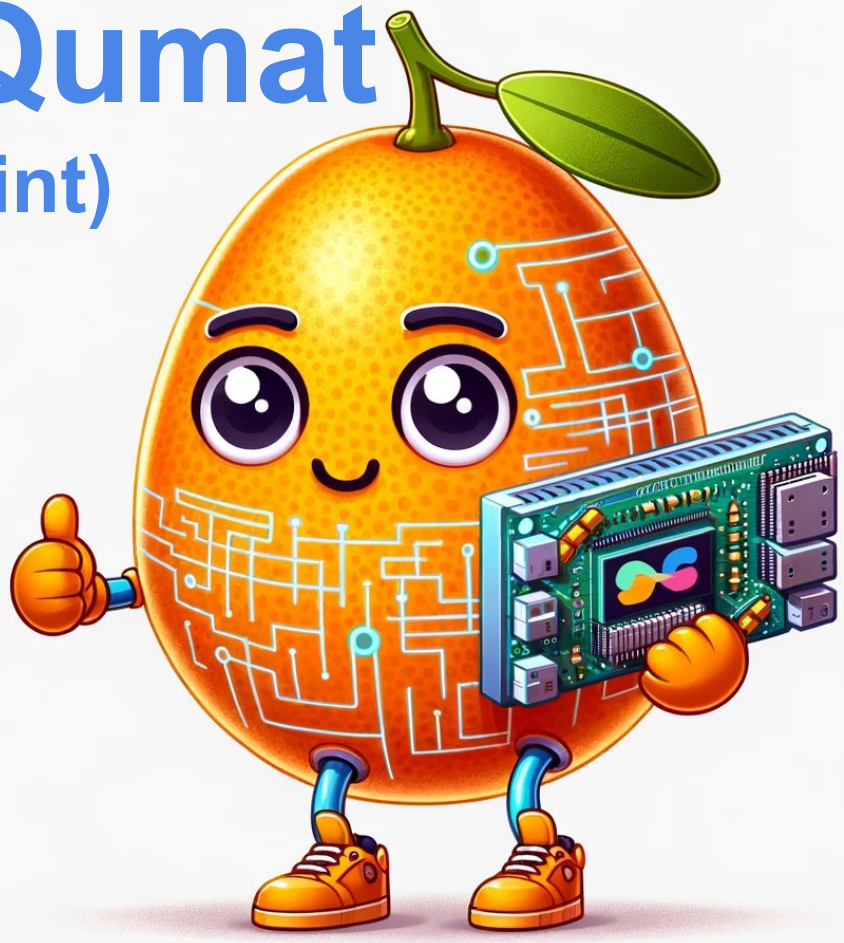
Introducing Qumat

(an Apache Mahout Joint)



FOSDEM 2025
Quantum Computing DevRoom
02 FEB 2025

Agenda

Who we are (and who we aren’t)

History of Apache Mahout

Why We Made Qumat

Current State of Qumat

Demo

Calls to Action / Q & A

Who We Are

Andrew Musselman

PMC Apache Mahout, wrote the book on Mahout for Springer

Head of technology at Speedchain, patents in computer vision and smart contracts

Two daughters, two dogs, two cars in Portland Oregon

Music, amateur radio, assorted hobbies

Trevor Grant

PMC Apache Mahout

Author-Kubeflow for Machine Learning (+Various papers on this/that/not quantum)

Father of Two Boys

Car free since ‘23 (actually ‘22, but it doesn’t rhyme)

A special thank you to IBM for its continued patronage of my artistic endeavors

Who We Aren’t

We are *NOT*

Quantum Researchers

Quantum PHDs

Authors of any notable papers on quantum this or that

WE ARE

Software engineers with a history and understanding of why it’s important to be able to abstract over multiple different backends.

History of Apache Mahout

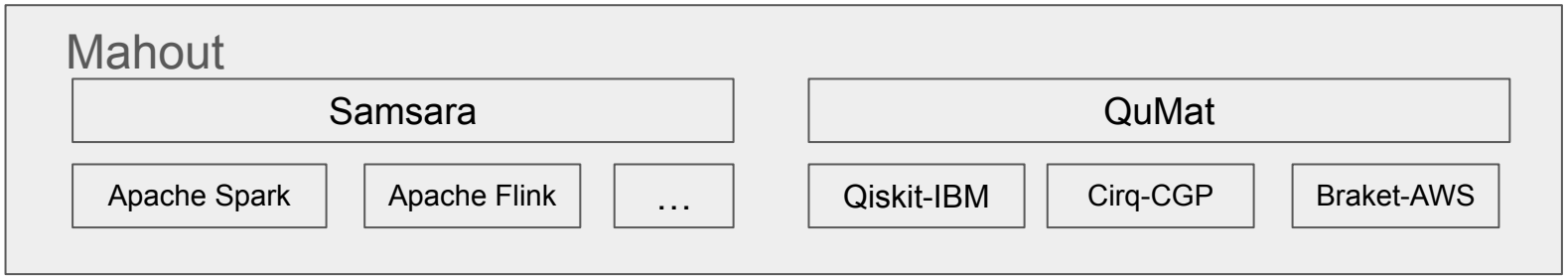
History of Mahout

- 2008: Apache Lucene sub-project with analytics including clustering
- 2010: Apache top-level project recommender system
- Taste: (https://svn.apache.org/repos/asf/lucene/mahout/site/publish/taste.pdf?p=836600)
- 2015: Mahout-Samsara (back-end independent programming environment)
- 2017: Apache Hadoop MapReduce back-end deprecated
- 2020: 14.1 released
- 2022: Apache Zeppelin integration and algorithm development framework
- 2024: Qumat quantum computing interface

Why We Made Qumat

Motivation for Quantum in Mahout

- Core capabilities of Mahout
  - Matrix arithmetic and operations
  - Linear algebra
  - Back-end agnostic, distributed or in-core CPU or GPU
  - Simplified syntax in matrix math DSL (Samsara)
  - Management of large-scale vectors and matrices
- Quantum compute in essence is matrix arithmetic
  - Circuits are composed by multiplying matrices
  - Vectors and matrices are complex-valued instead of binary or real



What the Math Looks Like

Quantum states are represented by "kets," from a notation known as bra–ket. The vector representation of a single qubit is

$$|a\rangle = v_0\,|0\rangle + v_1\,|1\rangle \rightarrow \begin{pmatrix} v_0 \\ v_1 \end{pmatrix}$$

where v_0 and v_1 are the complex probability amplitudes of the qubit, and the values zero and one are represented by the kets

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

* https://en.wikipedia.org/wiki/Quantum_logic_gate

What the Math Looks Like

$$|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \text{and} \quad |1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

These kets are basis vectors for a complex vector space. Logic gates in quantum computing are matrices which take actions on qubits.

For example, the identity gate (identity matrix) performs no action on any qubit:

$$\begin{pmatrix} v_0 \\ v_1 \end{pmatrix} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} v_0 * 1 + v_1 * 0 \\ v_0 * 0 + v_1 * 1 \end{pmatrix} = \begin{pmatrix} v_0 \\ v_1 \end{pmatrix}$$

Why we Need Qumat

- Multiple frameworks for programming quantum circuits
  - Qiskit (IBM)
  - Braket (AWS)
  - Cirq (Google)
  - TKET (Honeywell)
  - Qsharp (Microsoft)
  - Pyquil (Rigetti)
- Multiple vendor platforms for running on simulated or real quantum hardware
  - IBM (https://quantum.ibm.com)
  - AWS (https://aws.amazon.com/braket)
  - Google (https://quantumai.google/quantumcomputer)
  - Honeywell (https://www.honeywell.com/us/en/company/quantum/quantum-computer)
- Multiple frameworks and back-ends is a Mahout core value
- One interface with no code change across tools provides flexibility

ACKCHYUALLY - IBM’s Qiskit Already Does This

One Vendor controlling the abstraction layer is… problematic at best.

At FOSDEM I shouldn’t *need* to defend this.

Show of hands, do we need to explain this?

Current State of Qumat

Not Feature Complete (yet)

Basic Gates

- NOT
- Hammard
- C-NOT
- SWAP
- Pauli-X
- Pauli-Y
- Pauli-X

Not Feature Complete (yet)

Backends

- Qiskit
- Cirq
- Bracket

Not Feature Complete (yet)

Other

- Drawing Methods
- Parameterized Circuits
- Getting Started in Quantum Computing Guide
- Colab Runnable Notebook Examples

Roadmap / Future Work

Quantum Machine Learning

Implementing More Algorithms

Future FOSDEM Talks / Reasons to tell our job we’re going to Europe

VC Funding (?)

Demo

Demo

https://github.com/apache/mahout/blob/main/examples/Simple_Example.ipynb

https://colab.research.google.com/drive/14E36hXiOMmVek8uGhLUPJaXKterTVKVY?usp=sharing

Calls to Action / Q&A

Calls To Action

We’re not Quantum Experts, come check out code.

Quantum Machine Learning

Install with `pip install qumat` (https://pypi.org/project/qumat thank you `zonki`)

Play with Notebooks

Join the mailing list

Buy us beers (non-ethanol for Andrew)

Thank You