Quantum Walk Simulation on the QVM Architecture

Operator-Based Emulation of Quantum Walk Dynamics on Classical Distributed Systems

Honza Rožek

Abstract

This document specifies a method for simulating quantum walk dynamics on the Quantum Virtual Machine (QVM) architecture using classical, operator-based execution. The approach does not require quantum hardware and does not claim physical equivalence to quantum systems. Instead, it defines a deterministic, interference-based execution model that reproduces the structural behavior of quantum walks—superposition, interference, and controlled collapse—within the formal semantics of QVM.

1 Quantum Walk Simulation on the QVM Architecture

Motivation

Quantum walks are a fundamental computational primitive in quantum algorithms, with applications in search, sampling, and graph traversal. Their defining features include state superposition, interference between paths, and measurement-induced collapse.

Direct execution of quantum walks requires quantum hardware. However, many algorithmic advantages of quantum walks arise from their structural dynamics rather than from physical quantum effects. This motivates a classical, deterministic simulation framework capable of reproducing these dynamics at the level of state evolution and information propagation.

The QVM architecture provides such a framework by supporting operator-based state evolution with explicit interference and collapse semantics.

Scope and Non-Claims

This document:

- describes a simulation of quantum walk dynamics on classical hardware;
- defines a mapping between quantum walk components and QVM operators;
- specifies execution semantics suitable for distributed systems.

It explicitly does not claim:

- physical equivalence to quantum systems;
- exponential quantum speedup guarantees;
- simulation of entanglement at the physical level.

Quantum Walk Model

A discrete-time quantum walk is defined on a graph $G = (V, E)$ with state

$$|\Psi_t\rangle = \sum_{v \in V} \alpha_v^{(t)} |v\rangle,$$

where amplitudes evolve through alternating coin and shift operations, followed by optional measurement.

The essential structural elements are:

- distributed state over graph nodes;
- linear mixing of neighboring states;
- phase-sensitive interference;
- optional collapse or observation.

QVM State Representation

In QVM, the walk state is represented as a distributed collection of state components

$$\Psi_t = \{\psi_v^{(t)} : v \in V\},$$

where each component ψ_v corresponds to a graph vertex and contains:

- a bounded amplitude-like value;
- optional phase or orientation metadata;
- execution and provenance metadata.

State components are typed, non-addressable, and managed exclusively by the QVM runtime.

Operator Mapping

Quantum walk dynamics are mapped to QVM operators as follows:

- **Coin operation** $\rightarrow$ local modulation operator;
- **Shift operation** $\rightarrow$ interference (mixing) operator over graph neighborhoods;
- **Interference** $\rightarrow$ linear or nonlinear aggregation of neighboring states;
- **Measurement** $\rightarrow$ programmable collapse operator.

This mapping preserves the structural flow of information without requiring complex-valued quantum amplitudes or unitary constraints.

Execution Semantics

Each simulation step is defined as:

$$\Psi_{t+1} = \mathcal{C}_t \circ \mathcal{N}_t \circ \mathcal{U}_t \circ \mathcal{M}_t(\Psi_t),$$

where:

- $\mathcal{M}$ encodes coin-like modulation;
- $\mathcal{U}$ propagates state across graph edges;
- $\mathcal{N}$ enforces normalization and stability;
- $\mathcal{C}$ performs optional observation or pruning.

All operators are deterministic given fixed parameters and seeds.

Distributed Execution

Graph partitions may be distributed across execution nodes. Interference operators exchange boundary state according to explicit topology rules. Asynchronous execution is permitted under bounded staleness assumptions, with stability ensured via damping and normalization.

Applications

Simulated quantum walks on QVM are applicable to:

- graph exploration and ranking;
- search heuristics and optimization;
- diffusion-based learning and inference;
- distributed decision and control systems.

The framework is particularly suitable where determinism, auditability, and classical deployability are required.

Limitations

The simulation does not preserve quantum unitarity or entanglement. Interference is algorithmic rather than physical. Performance characteristics differ from true quantum execution and depend on operator configuration and graph structure.

Conclusion

The QVM architecture enables a principled simulation of quantum walk dynamics on classical distributed systems. By expressing superposition, interference, and collapse as explicit operators, the framework captures the algorithmic essence of quantum walks while remaining deterministic, verifiable, and deployable on existing infrastructure.

This approach provides a bridge between quantum-inspired algorithms and practical classical execution environments.