Quantum Computing (Conceptual) and Compute Architectures

Note on tooling. For this module, students are **not expected to run** the quantum computing code in class due to environment and dependency constraints. Instead, you will **read** the provided annotated Python script and answer questions that require careful reasoning about the mathematics and the compute implications.

Reading. `quantum_computing.py` (non-executable teaching script).

Conceptual Questions

Please write three to ten sentence explanations for each of the following questions. **You are only required to answer ONE of the two questions below.**

1. Quantum state-vector simulation scales exponentially with the number of qubits. Explain why this creates a **parallel computing** problem. Then compare three compute options for accelerating large simulations: (i) multi-core CPU parallelism, (ii) GPUs, and (iii) TPUs. In your comparison, discuss (a) what parts of the computation parallelize well, (b) the memory bottleneck, and (c) one scenario where each option is likely to be preferred.
2. Many research pipelines mix different compute workloads (hyperparameter sweeps, Monte Carlo, deep learning, large matrix operations). Explain how you would decide whether to use **CPU parallelism**, **GPU acceleration**, or a **TPU** for a given task. In your answer, give one example of a workload that is a good fit for each (CPU, GPU, TPU), and explain one common mistake researchers make when choosing hardware (e.g., ignoring data movement/I/O, not accounting for batch size, assuming "GPU = always faster").

Applied Exercises

Use the code in the week's code tutorial and the lecture slides to answer the following questions.

3. **Single-qubit reasoning (state vectors and gates).** Using the definitions in `quantum_computing.py`:
   - Write the computational basis vectors $|0\rangle$ and $|1\rangle$.
   - Let $|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$. Derive expressions for:
     (a) $X|\psi\rangle$ (NOT gate),
     (b) $Z|\psi\rangle$ (phase flip),
     (c) $H|0\rangle$ and the measurement probabilities of the outcome (0 vs 1).
   - In 3–6 sentences, explain (in words) why applying H to $|0\rangle$ leads to a 50/50 measurement outcome even though the transformation is deterministic.
4. **Two-qubit reasoning (tensor products and entanglement).** Using the script's Bell-state construction:

- Show the algebraic steps that transform $|00\rangle$ into:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$$

  by applying (i) H on the first qubit and then (ii) CNOT (control=first, target=second).
- Using measurement logic, state:
  (a) $P(00)$, $P(01)$, $P(10)$, and $P(11)$ when measuring $|\Phi^+\rangle$ in the computational basis;
  (b) what you can infer about the second qubit after measuring the first qubit.
- In 4–8 sentences, explain what makes this state *entangled* (i.e., why it cannot be written as $|a\rangle \otimes |b\rangle$).

5. **Compute implications: exponential growth, memory, and "shots."** This question connects the conceptual script to real compute constraints.
   - State-vector size: a pure state over n qubits requires 2^n complex amplitudes. Assume complex numbers are stored as complex64 (8 bytes real + 8 bytes imag = 16 bytes).
     (a) Compute the approximate memory required (in GB) to store the full state vector for $n = 25$, $n = 30$, and $n = 35$ qubits.
     (b) Briefly interpret what this implies for laptop vs workstation vs cluster computing.
   - "Shots": In the Qiskit example, the circuit is run for many shots (e.g., 1000).
     (a) Explain why the observed counts are close to (but not exactly) 50/50.
     (b) Explain one reason why increasing shots improves estimation of probabilities but does *not* remove bias due to noise on real hardware.
   - In 5–8 sentences, propose one parallelization strategy for a quantum workflow:
     (a) either parallelize over **shots** (embarrassingly parallel sampling),
     (b) or parallelize the **state-vector** computation (harder; memory-bound),
     (c) and state whether you would prefer CPU parallelism, GPU, or TPU for your chosen strategy and why.
6. **Challenge Question (Optional — if you finish early):** Choose **ONE** option:
   (a) **Bell states extension.** Write the algebra (and/or circuit description) to construct at least **two** additional Bell states (e.g., Φ^- and Ψ^+). For each, state the measurement outcome support (which bitstrings can appear) and the associated probabilities.
   (b) **Compute plan memo.** Write a short (8–12 sentence) "compute plan" for running a larger simulation study (e.g., a sweep over circuit depths and noise levels). Your memo must mention: (i) how you would use parallel computing (what is parallelized), (ii) whether you would use CPU, GPU, or TPU and why, and (iii) one reproducibility step you would take (seeds, environment capture, logging, artifact saving).