FSDC

Quantum Counting
FSDC

Bachelor of Technology

in

Information Technology

by

Your Name

SCHOOL OF ENGINEERING
AJEENKYA D Y PATIL UNIVERSITY

Abstract

The advent of quantum computing is poised to revolutionize various sectors, including cloud computing. This paper explores the hybrid integration of quantum and classical cloud systems, focusing on the project *Quantum-Cloud Integration*. The project demonstrates the practical implementation of hybrid cloud-quantum systems, leveraging tools like Docker, IBM Quantum, and AWS to showcase how quantum capabilities can enhance traditional cloud infrastructure. This research highlights the framework, challenges, and potential impacts on cloud computing paradigms.

1 Introduction

Quantum computing represents a significant leap forward from classical computing, offering unparalleled computational power for specific problems. Cloud computing, on the other hand, provides scalable and accessible computational resources. The integration of these technologies can lead to innovative solutions for industries requiring high-performance computing.

1.1 Objectives

This paper aims to:

- Examine the *Quantum-Cloud Integration* project.
- Discuss the tools and methodologies employed.
- Analyze the challenges and potential benefits of hybrid quantum-cloud systems.

2 Methodology

2.1 Framework Design

The integration framework is built using Docker containers to manage classical cloud workloads and IBM Quantum for quantum computations. A layered approach ensures seamless interaction between quantum resources and classical systems.

2.2 Tools and Technologies

- **Docker:** Containerization of cloud applications for easy deployment.
- **IBM Quantum:** Access to quantum algorithms and processing power.
- **AWS:** Classical cloud resources used for scalable storage and computational tasks.
- **Python and Qiskit:** Programming quantum algorithms and orchestration.

2.3 Workflow

Below is an example quantum workflow implemented in Python:

Quantum Circuit Example

```python
from qiskit import QuantumCircuit, Aer, execute

# Create a simple quantum circuit
qc = QuantumCircuit(2, 2)
qc.h(0)  # Apply Hadamard gate
qc.cx(0, 1)  # Apply CNOT gate
qc.measure([0, 1], [0, 1])

# Simulate the circuit on Aer simulator
simulator = Aer.get_backend('qasm_simulator')
result = execute(qc, simulator, shots=1024).result()

# Get and display results
counts = result.get_counts()
print("Quantum Results:", counts)
```

3 Implementation

3.1 Docker Configuration

The Docker setup ensures an isolated environment for managing hybrid operations. Below is a snippet of the 'Dockerfile':

Dockerfile Example

```dockerfile
# Use an official Python runtime as the base image
FROM python:3.9-slim

# Set the working directory
WORKDIR /app

# Copy project requirements
COPY requirements.txt ./

# Install dependencies
RUN pip install --no-cache-dir -r requirements.txt

# Copy the rest of the application code
COPY . .

# Define the command to run the application
CMD ["python", "main.py"]
```