ETH Quantum Hackathon 2024

NVIDIA Challenge:

Distributed Quantum Computing and the Vehicle Routing Problem

Challenge Goals

- Create a novel and useful application for the Vehicle Routing Problem (VRP)
- Translate the VRP to an instance of the Max Cut Problem
- Implement QAOA on a GPU using CUDA-Q to solve a small Max Cut Problem
- Investigate options including circuit cutting and distributed computing to scale QAOA to handle larger max cut problems.
- Relate your solution to the max cut problem back to the original VRP and communicate your work effectively.

Context

The Vehicle Routing Problem (VRP) addresses the question: "what is the optimal set of routes for a fleet of vehicles to deliver to a given set of customers?". One can think about the VRP as a generalization of the travelling salesman problem (TSP). VRP has applications in many sectors, especially in logistics and transportation.

The max cut problem is a combinatorial optimization problem that asks for the maximum number of edges that can be cut by partitioning the nodes of a graph into two disjoint subsets.

CUDA-Q

NVIDIA CUDA-Q is a unified programming model designed for a hybrid setting, which allows integrating and programming quantum processing units (QPUs), GPUs, and CPUs in one system. It is an open-source platform and it consists of language extensions for Python and C++ and a system-level toolchain that enables application acceleration.

CUDA-Q enables GPU-accelerated system scalability and performance across heterogeneous QPU, CPU, GPU, and emulated quantum system elements.

Tasks

Task 1: Choose your application

Describe your chosen application: state the problem you want to solve, how it relates to the VRP problem, variables you will use, relevance and industries/sectors that could benefit from the application.

Task 2: Formulate the solution

Explain your reasoning for your solution. Be clear and organized, writing equations when needed. Don’t forget to declare the variables you are using. Diagrams, graphics and drawings are encouraged. You may opt to apply the clustering techniques described in Feld et al and Herzog et al to translate the CVRP to a Max Cut Problem.

Task 3: Implement QAOA to solve a small problem using CUDA-Q

Develop the code to implement your solution using CUDA-Q, the platform for hybrid quantum-classical computing. To set up CUDA-Q on your local machine, check out the Quick Start Guide. You will also have access to GPUs through Denvr Dataworks. You'll be provided URLs and instructions for logging into Denvr Dataworks in a separate correspondence.

Here are some useful resources:

- A Review of QAOA and its Variants by Blekos et al
- QAOA example in CUDA-Q.
- CUDA-Q documentation.
- CUDA-Q GitHub

Task 4: Run and scale your problem to larger sizes using techniques such as circuit cutting and distributed computing

The Blekos et al review of QAOA examines several variations of QAOA. Create your own variation or select from the variations discussed in the review, and implement the variation in QAOA. Determine which is the largest graph that you can solve with your variation. Verify your solutions classically.

If you aren't sure where to start, take a look at the divide-and-conquer variation that has been explored in many papers including Zhao et al. The divide and conquer approach to QAOA for max cut is an example of circuit cutting which allows for parallel simulation of smaller quantum circuits, which could help scale the algorithm to larger problem sizes.

Task 5: Present your results

Summarize your results in a document (single .pdf file) which includes a description of the problem you want to solve and relationship to Max-Cut, your solution, your implementation and the results. Provide your code in a repository or file.