Georgia Tech | Center for Research into Novel Computing Hierarchies

Enabling a Programming Environment for an Experimental Ion Trap Quantum Testbed

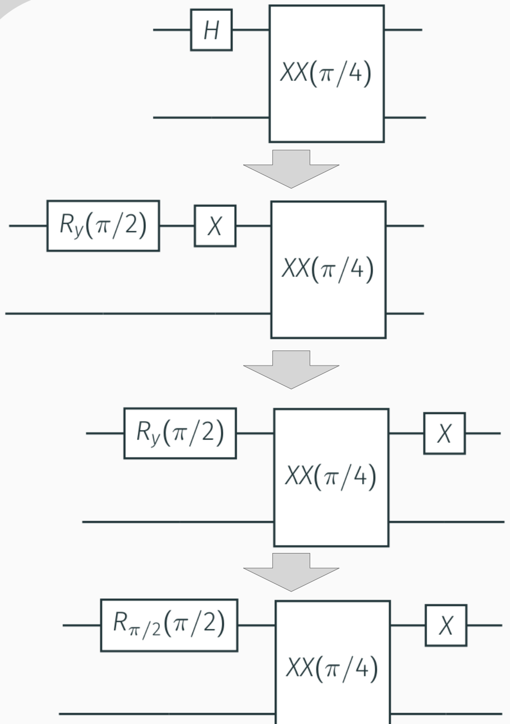
Austin Adams[1], Elton Pinto[1], Jeff Young[1], Creston Herold[2], Alex McCaskey[3], Eugene Dumitrescu[3], Tom Conte[1]

School of Computer Science [1], GTRI [2], ORNL [3]

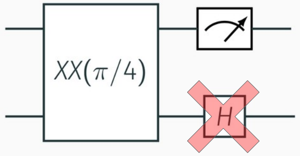
ICRC ‘21
arXiv:2111.00146

Single-Qubit Gate Compiler Pass

Example of a compiler optimization performed: Commuting X-rotations around native XX gates (a form of Mølmer–Sørenson gate)

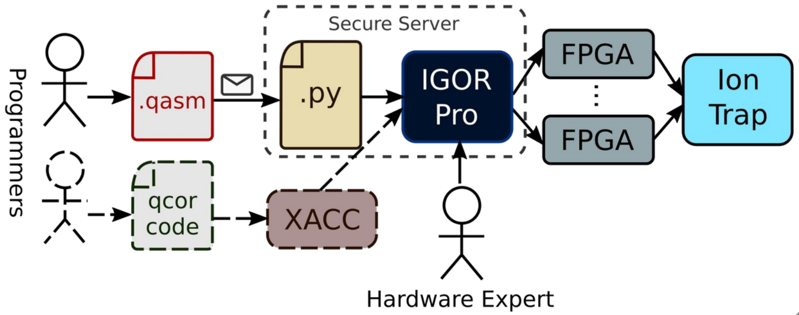


Another example: Quantum dead code elimination



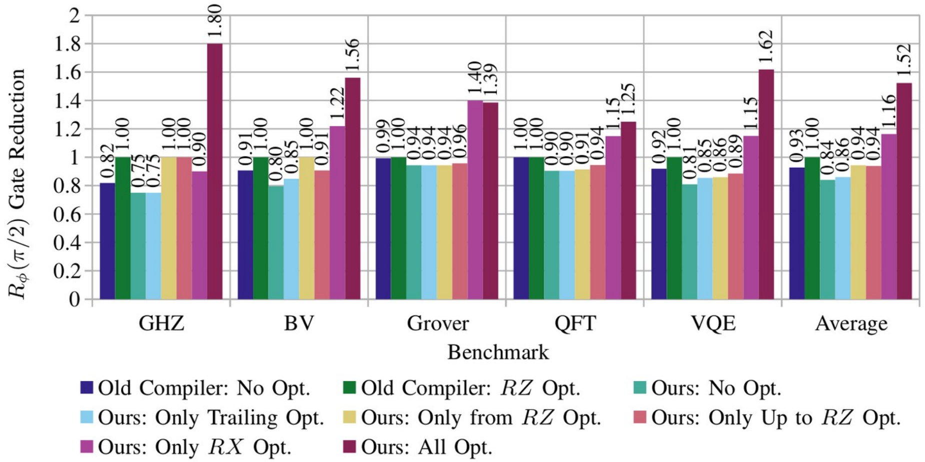
Goal: Implement a Programmer-Directed Flow for Non–Hardware Experts to Program an Ion Trap Quantum Computer at GTRI

We added the dashed path shown below, starting from the bottom left:



Evaluation Results

With our benchmarks, we saw an average of 1.52x reduction in native single-qubit gates compared to the existing compiler:



QCOR/XACC

We implemented a backend for QCOR/XACC, which allows programmers to write CUDA-like heterogeneous quantum–classical programs for the GTRI testbed:

```
__qpu__ void ghz(qreg q) {
  H(q[0]);
  for (int i = 1; i < q.size(); i++)
    CNOT(q[i-1], q[i]);
  Measure(q);
}

int main(int argc, char **argv) {
  auto q = qalloc(atoi(argv[1]));
  ghz(q);
  q.print();
}
```

Future Work

These optimizations are made at runtime. What about compile-time quantum–classical optimizations?

```
Rx(π/4)
if input == 42 {
  Rx(π/4)
}
```

Input=42: 8 native gates
Input≠42: 4 native gates

```
if input == 42 {
  Rx(π/2)
} else {
  Rx(π/4)
}
```

Input=42: 1 native gate
Input≠42: 4 native gates

Georgia Tech