PHOENIX: Pauli-based High-level Optimization Engine for Instruction Execution on NISQ Devices

Algorithm 1: Pauli Strings Simplification in BSF

Input : Pauli strings list *pls*
Output: Reconfigured circuit components list *cfg*

$cfg \leftarrow \emptyset$; $bsf \leftarrow$ BSF(*pls*); $cliffs\_with\_locals \leftarrow \emptyset$;
while *bsf*.TOTALWEIGHT() > 2 **do**
  $local\_bsf \leftarrow bsf$.POPLOCALPAULIS();
  $C \leftarrow \emptyset$; // Clifford2Q candidates
  $B \leftarrow \emptyset$; // Each element of B results from applying each Clifford2Q candidate on *bsf*
  costs $\leftarrow \emptyset$; // Cost functions calculated on each element of B
  for *cg* ***in*** CLIFFORD_2Q_SET **do**
    for *i, j* ***in*** COMBINATIONS(RANGE(n), 2) **do**
      cliff $\leftarrow cg$.ON(i, j) ; // qubits acted on
      $bsf' \leftarrow bsf$.APPLYCLIFFORD2Q(*cliff*);
      cost $\leftarrow$ CALCULATEBSFCOST(bsf');
      C.APPEND(*cliff*);
      B.APPEND(bsf');
      costs.APPEND(*cost*);
    end
  end
  $bsf \leftarrow$ BSFWITHMINCOST(B, *costs*);
  cliff $\leftarrow$ CLIFFORDWITHMINCOST(C, *costs*);
  cliffs_with_locals.APPEND((*cliff*, *local_bsf*));
end
cfg.APPEND(*bsf*);
for *cliff, local_bsf* ***in*** *cliffs_with_locals* **do**
  // Clifford2Q operators are added as conjugations, with local Pauli strings peeled before each epoch
  cfg.PREPEND(*cliff*);
  cfg.APPEND(*local_bsf*);
  cfg.APPEND(*cliff*);
end

***Abstract*—Quantum computing ...**

We propose a Pauli-based High-level Optimization ENgine for Instruction eXecution (PHOENIX) of Hamiltonian simulation programs on NISQ devices

I. INTRODUCTION

Quantum computing ...

II. MOTIVATION

[ZY: Motivation and preliminary knowledge]

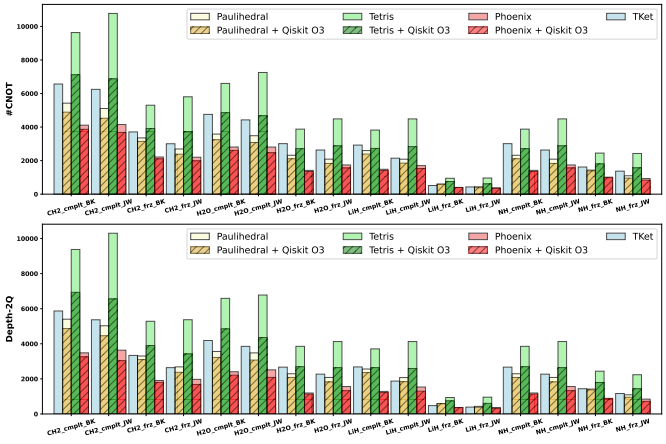

Fig. 1. Bechmarking on logical-level synthesis (all2all topology)

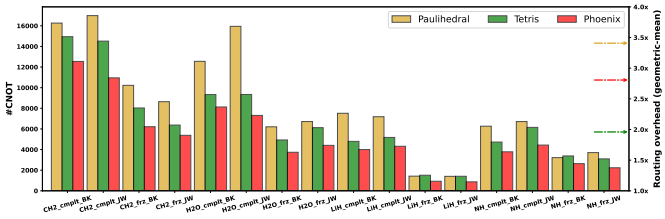

Fig. 2. Hardware-aware compilation for limited-topology NISQ device

III. OUR PROPSAL: PHOENIX

A. Overall framework

B. BSF simplification for each IR group

C. Ordering of IR groups

IV. EVALUATION

A. Experimental settings

B. Benchmarks

1) Metrics:

2) Baselines:

- TKET
- PAULIHEDRAL
- TETRIS

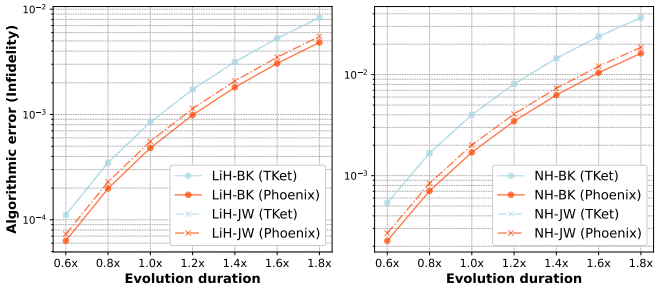

Fig. 3. Algorithmic error

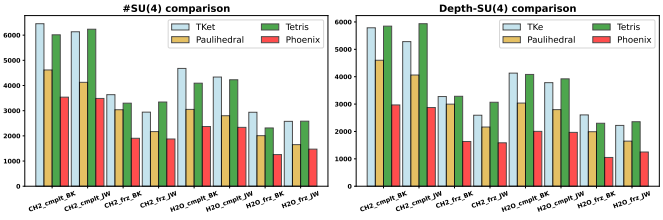

Fig. 4. SU(4) ISA comparison

C. Logical-level compilation

D. Hardware-aware compilation

E. Diverse ISA comparison

F. Breakdown analysis

G. Real system evaluation

H. Scalability