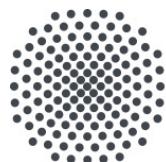


Quantum Hardware Selection



University of Stuttgart

Institute of Architecture of Application Systems

Marie Salm

marie.salm@iaas.uni-stuttgart.de

Supported by:



Federal Ministry
for Economic Affairs
and Climate Action

on the basis of a decision
by the German Bundestag

About Me

- Marie Salm, 28
- B.Sc. & M.Sc. in computer science
- Ph.D. candidate & research associate
 - @ IAAS of Frank Leymann, University of Stuttgart
 - Since December 2019
 - Project PlanQK 
- Research topic: NISQ Analyzer (*see next slides*)



<https://www.iaas.uni-stuttgart.de/en/institute/team/Salm/>

©Marie Salm 2

Tutorial Structure

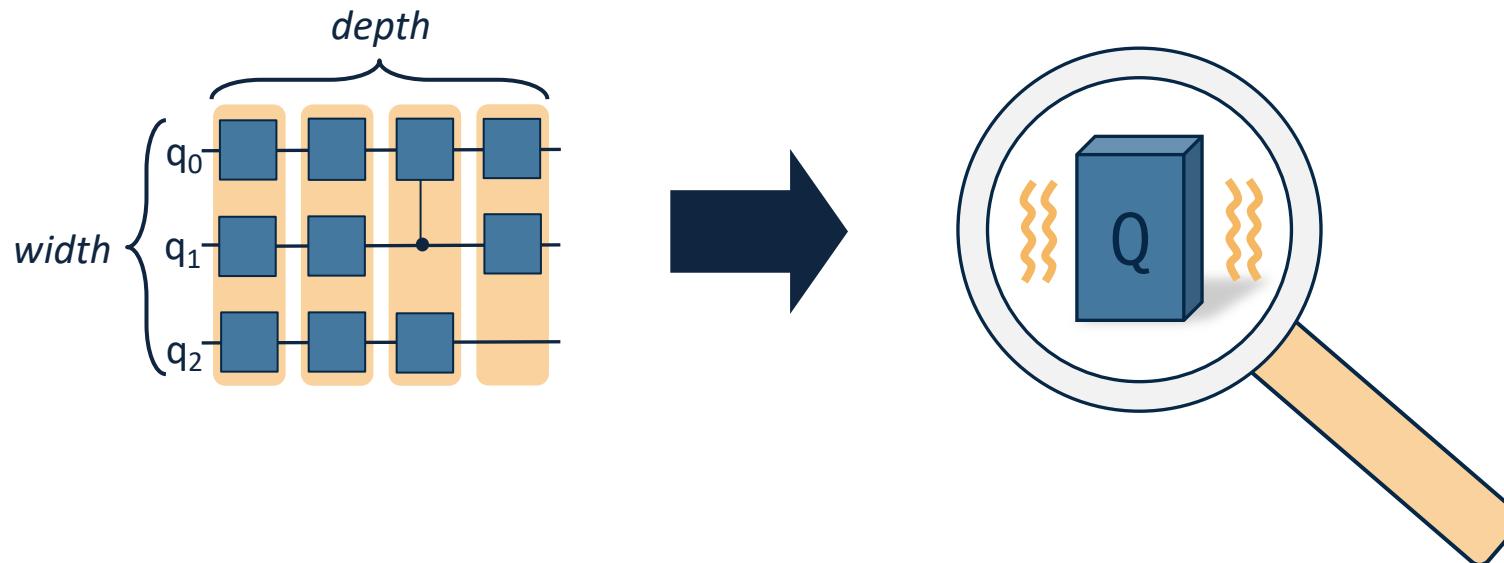
- Session 1 (09:00 - 10:30): An Introduction to Quantum Computing
- **Session 2 (11:00 - 12:30): Quantum Software Engineering**
 - Quantum software development lifecycle
 - Quantum hardware selection
 - Q/A session
 - Outlook to the afternoon sessions
- Session 3 (14:00 - 15:30): Quantum Workflows
- Session 4 (16:00 - 17:30): Operation of Hybrid Quantum Applications

Tutorial Structure

- Session 1 (09:00 - 10:30): An Introduction to Quantum Computing
- **Session 2 (11:00 - 12:30): Quantum Software Engineering**
 - Quantum software development lifecycle
 - **Quantum hardware selection**
 - Q/A session
 - Outlook to the afternoon sessions
- Session 3 (14:00 - 15:30): Quantum Workflows
- Session 4 (16:00 - 17:30): Operation of Hybrid Quantum Applications

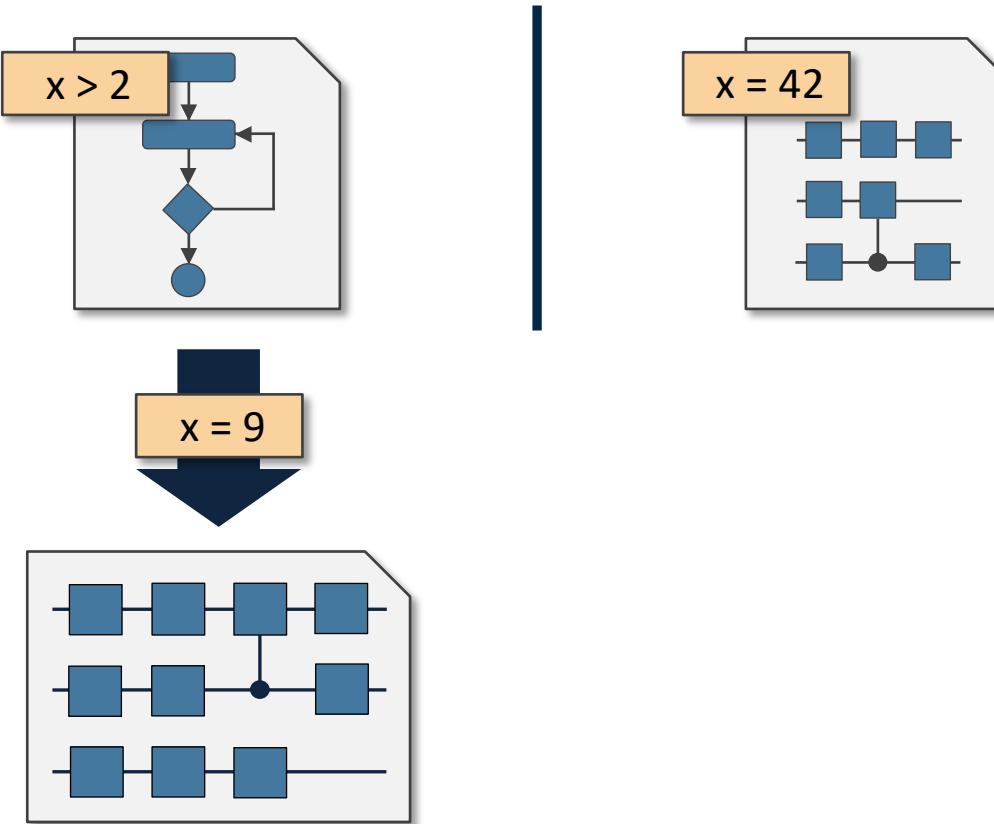
The NISQ Problem

- Currently in **Noisy Intermediate-Scale Quantum** era
 - Small number of qubits, high error rates
 - Limits width & depth of quantum circuits
 - Design circuits as **resource efficient** as possible



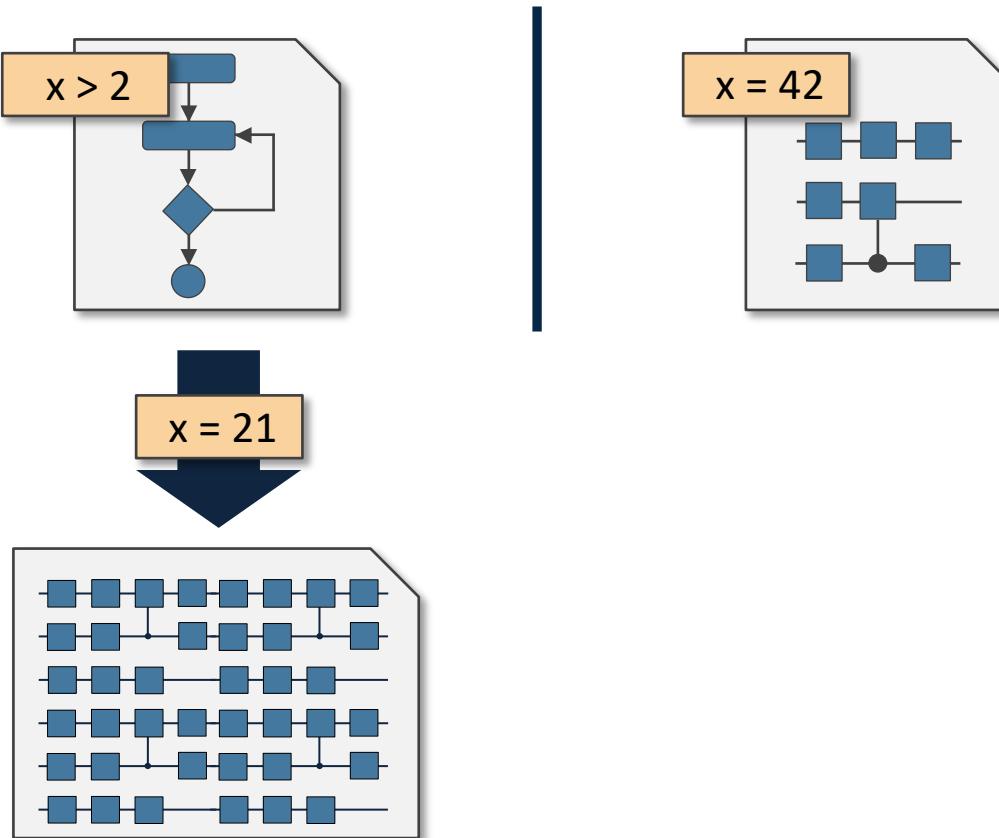
Quantum Implementations

- Varying **implementations** of quantum algorithms
 - Restrictions in input values



Quantum Implementations

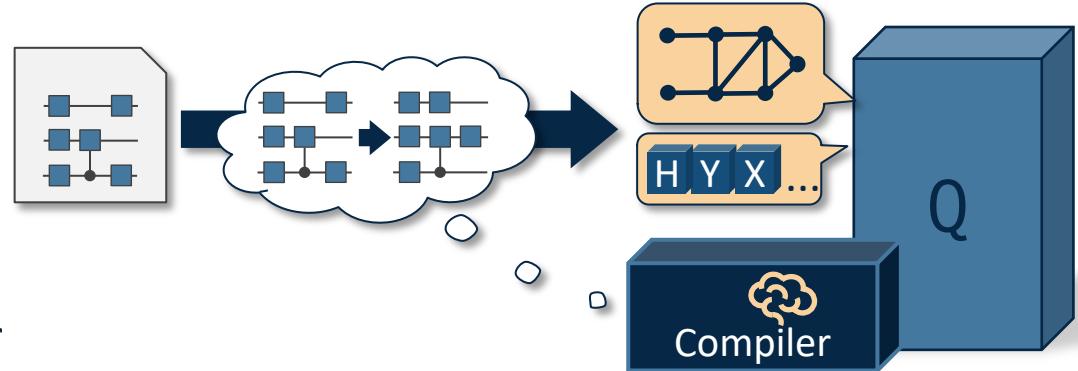
- Varying **implementations** of quantum algorithms
 - Restrictions in input values



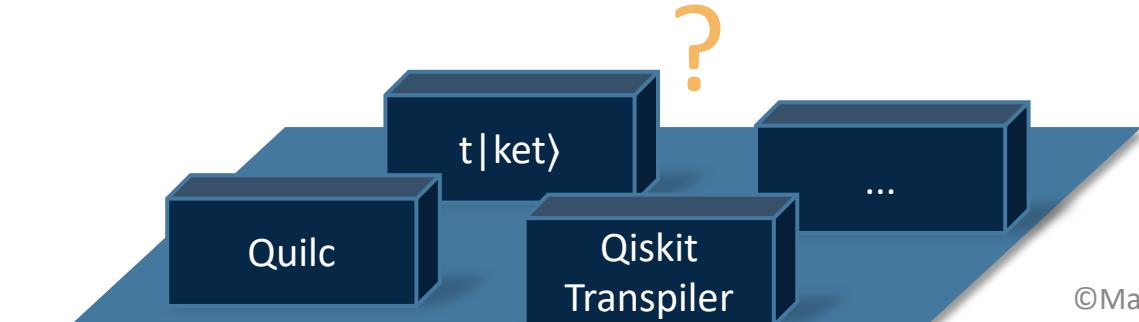
- Resulting **circuit size** depends on **input**

Influencing Circuit Size

- **True size of circuit depends on**
 - Qubit topology
 - Implemented gate set
 - Mapping & optimization by compiler

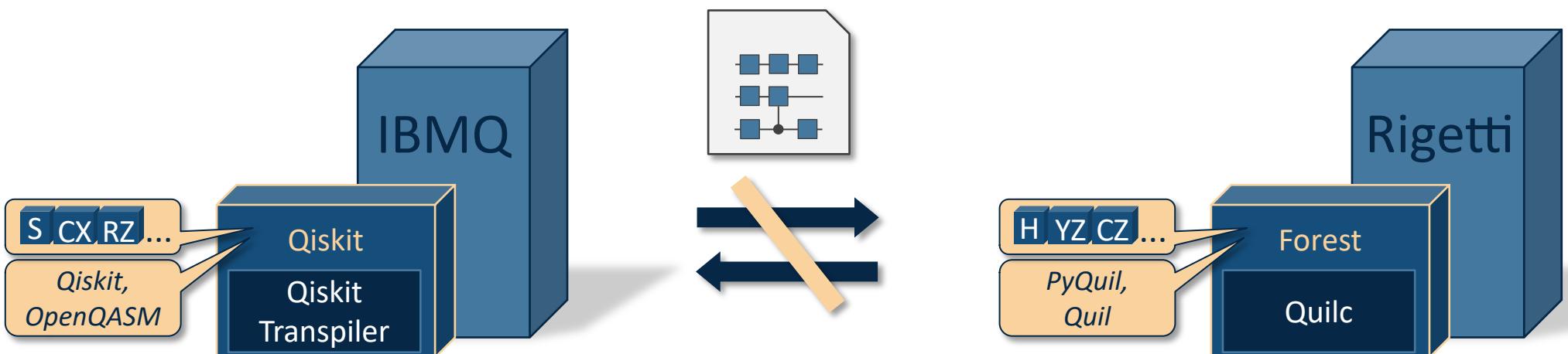


- **Variety** of quantum compilers exists
- Given a circuit: which compiler delivers “**best**” compilation result?
- Comparison is required ...



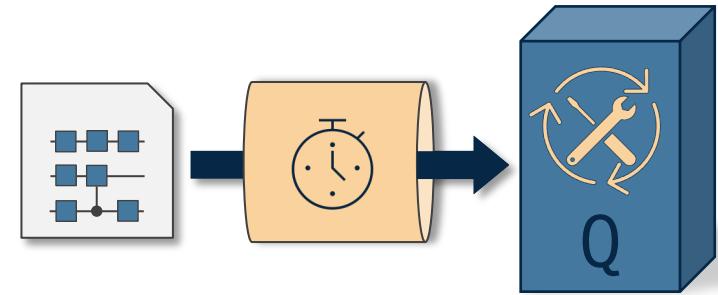
Usage of Quantum Compilers

- Most compilers **accessible via SDKs** differing in
 - Gate sets
 - Programming languages
 - Vendors of QPUs



In Addition ...

- QPUs are regularly re-calibrated
 - Changes in error rates
- Cloud providers regulate computing access
 - Resulting in waiting times
- Execution of quantum algorithm
 - Implementation
 - SDK
 - QPU
 - Compiler
 - Access regulations

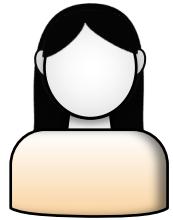


*“How can the **selection** of quantum algorithm **implementations** & **suitable quantum resources** be **automated** based on the **input data** of the chosen algorithm & the **user’s needs**?“*

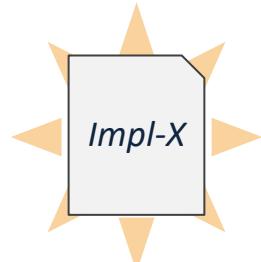
The Vision

Vision

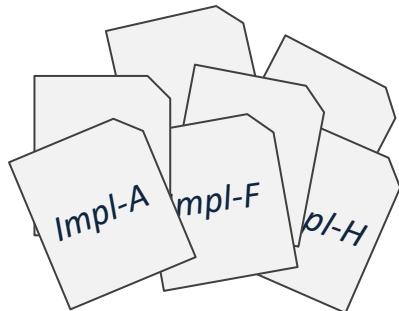
Factorizing



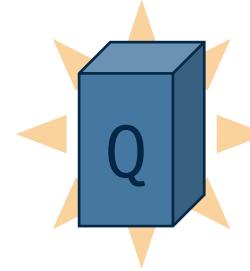
$N = 144$
Shor!



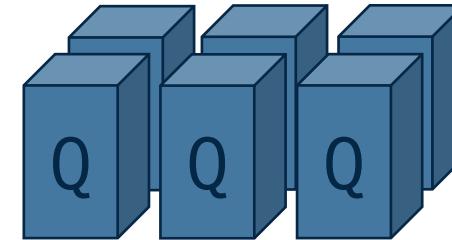
✓ 144



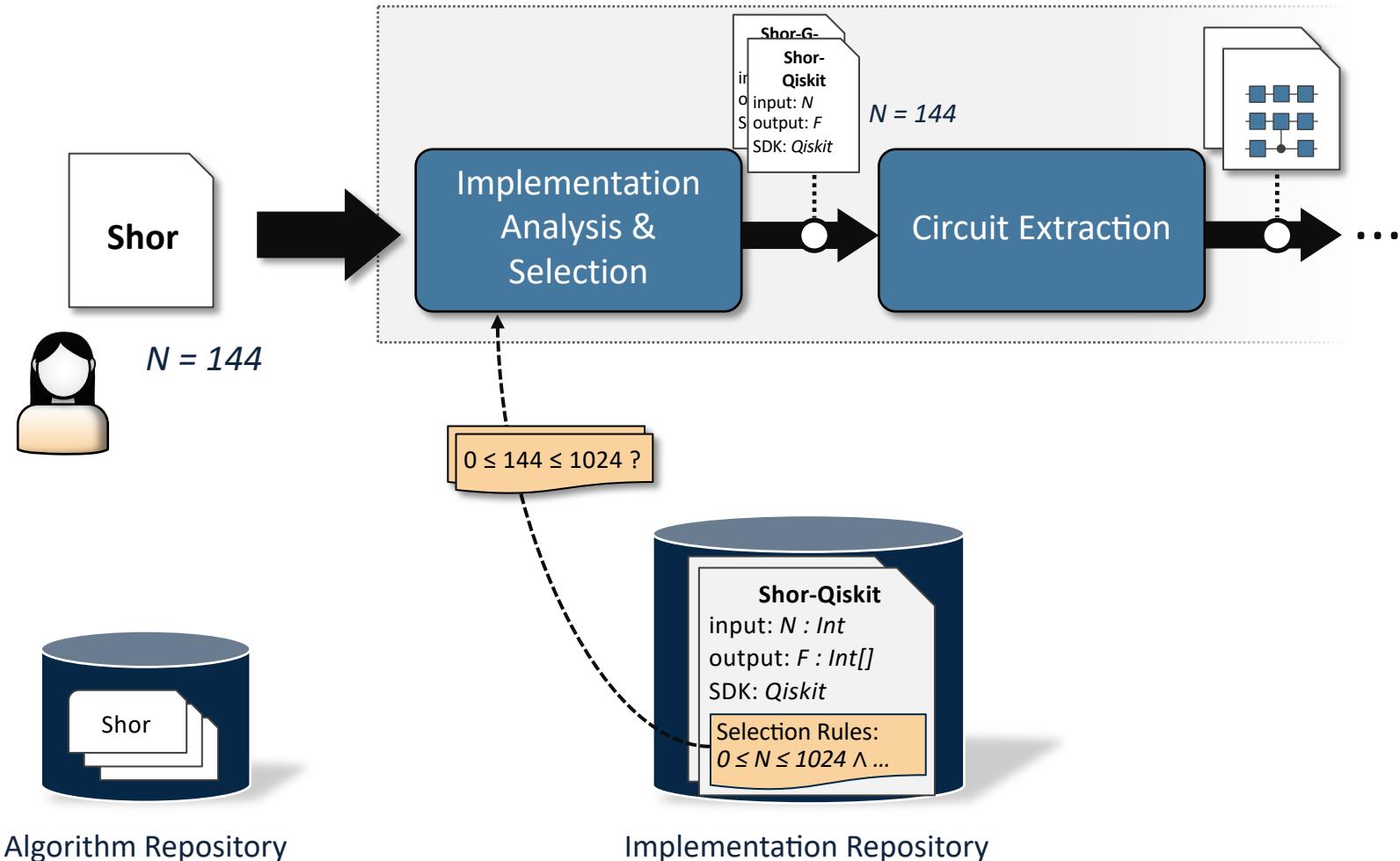
+



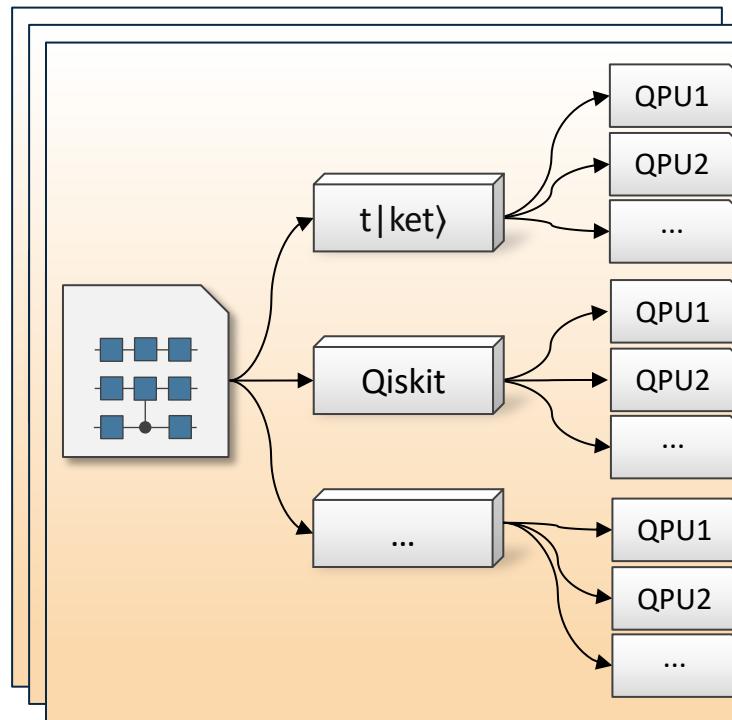
enough
✓ *qubits*
✓ *decoherence time*
✓ ...



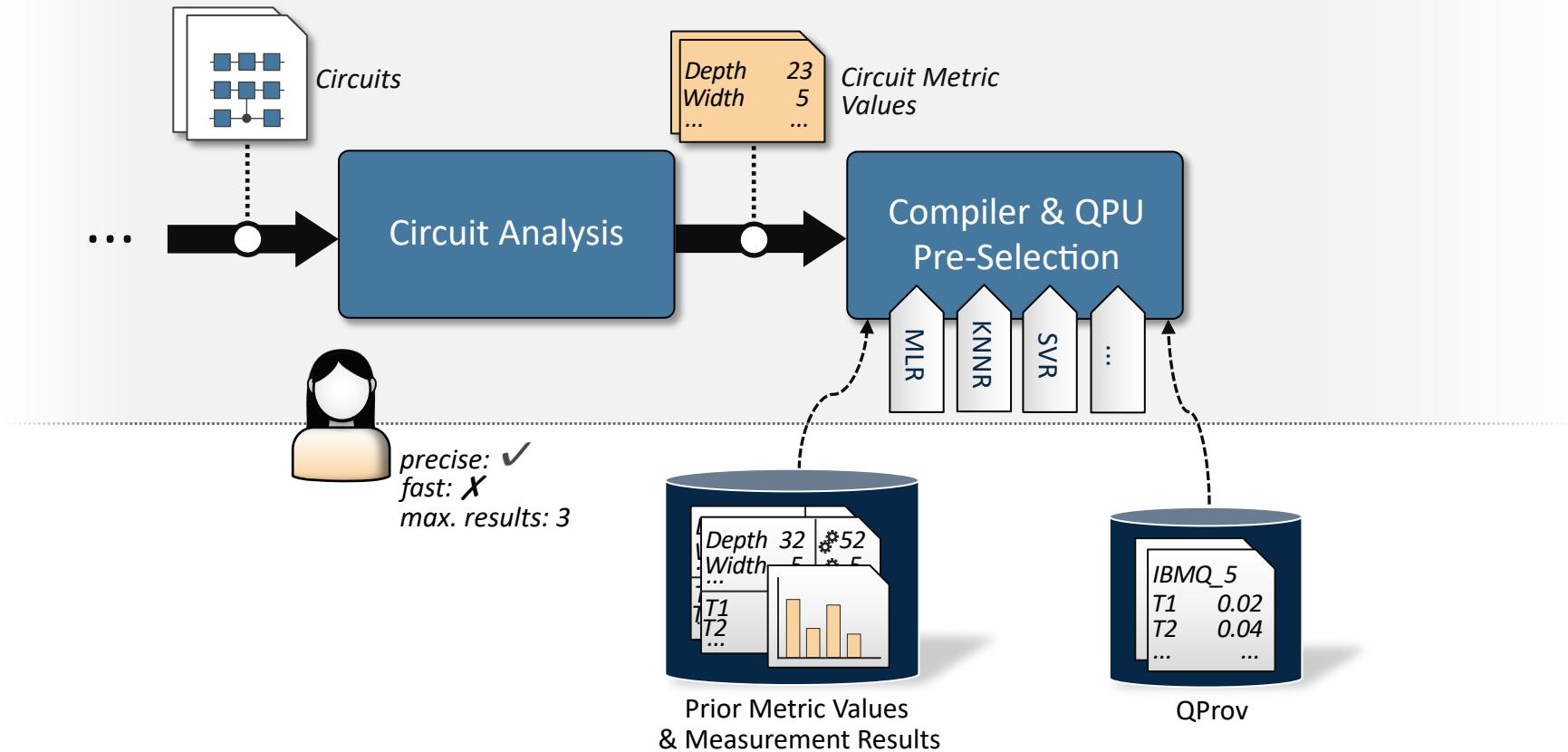
Selection of Quantum Implementations



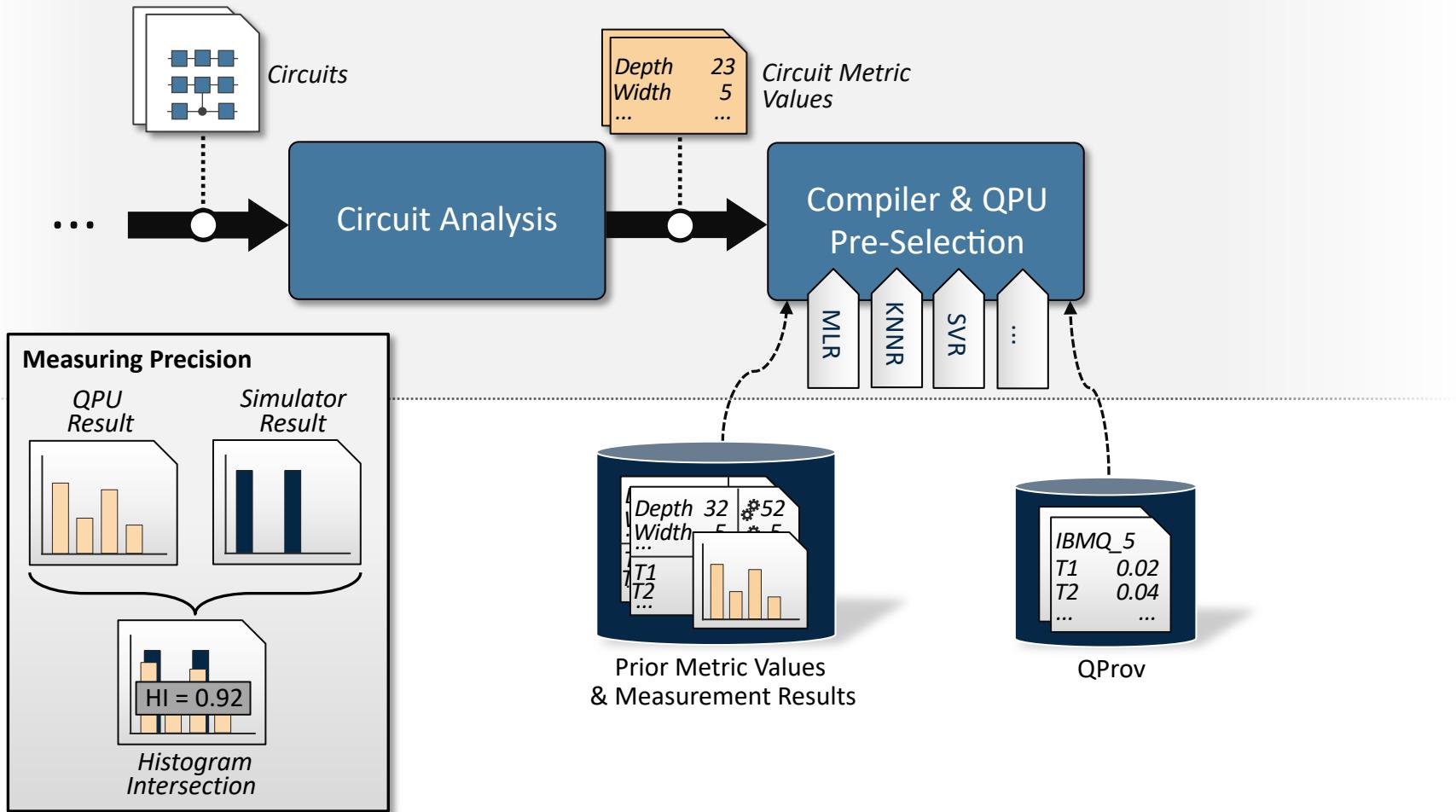
Estimation of Quantum Resources



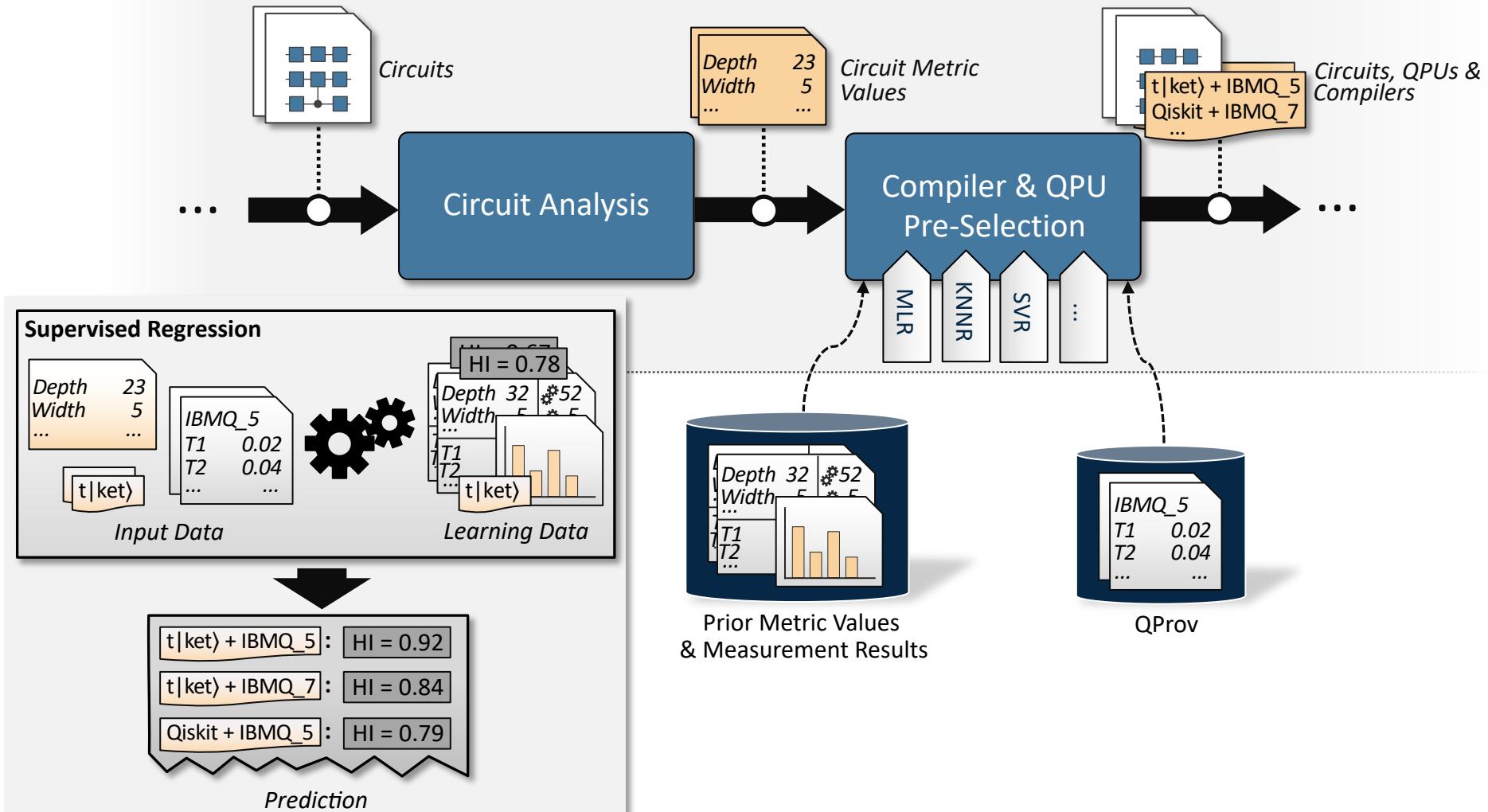
Estimation of Quantum Resources



Estimation of Quantum Resources



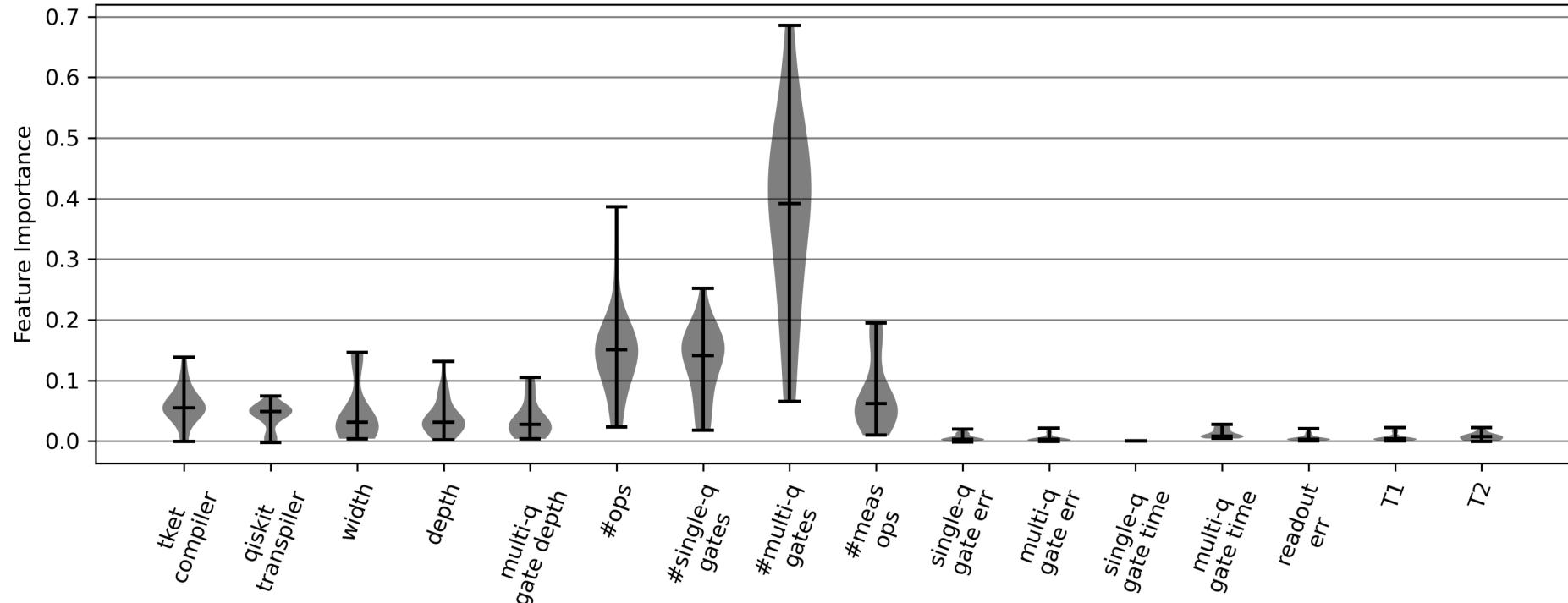
Estimation of Quantum Resources



Case Study – Setup

- IBMQ 5-qubit QPUs & simulator
- Qiskit transpiler & $t|ket\rangle$ compiler
- 52 input circuits
 - 3 algorithmic circuits
 - randomized circuits with Clifford gates
- 229 compiled & executed circuits
- support multiple variants of ML algorithms
 - decision trees
 - SVR
 - KNN regression
 - multiple linear regression

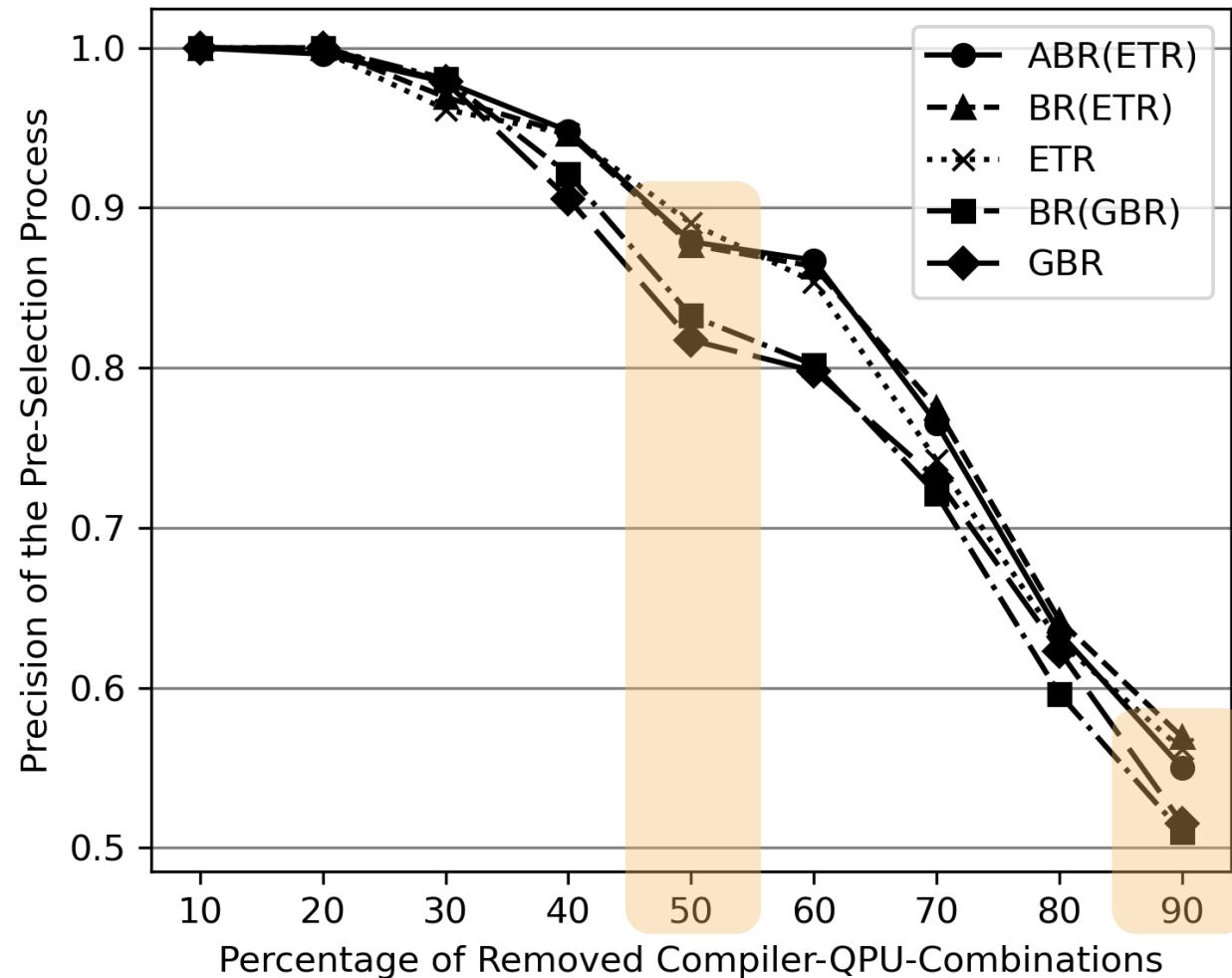
Case Study: Influence of Metrics



Case Study: Precision of Pre-Selection

meta estimators:
ABR: Ada Boost Regressor
BR: Bagging Regressor

regression algorithms:
ETR: ExtraTreesRegressor
GBR: Gradient Boosting Regressor



Case Study: Runtime Analysis

Table 1: Median runtimes of our approach removing various percentages of compiler-QPU-combinations (in seconds).

Circuit	0%	50%	70%	90%
Gr3	84.34	51.72	42.01	31.56
RC4	108.80	63.05	51.49	39.32
RC5	109.81	65.65	54.39	42.88



40% faster

Case Study: Runtime Analysis

Table 1: Median runtimes of our approach removing various percentages of compiler-QPU-combinations (in seconds).

Circuit	0%	50%	70%	90%
Gr3	84.34	51.72	42.01	31.56
RC4	108.80	63.05	51.49	39.32
RC5	109.81	65.65	54.39	42.88

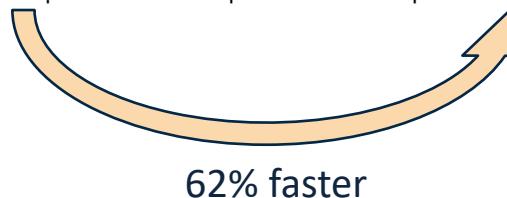


51% faster

Case Study: Runtime Analysis

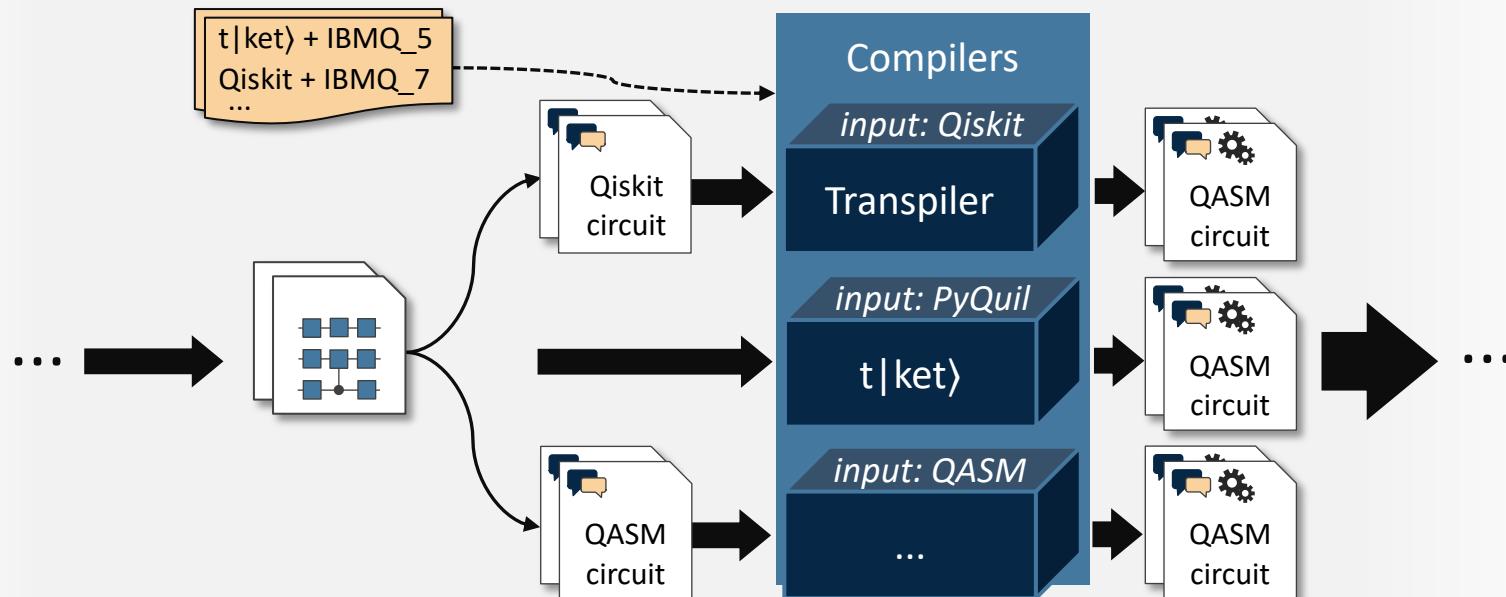
Table 1: Median runtimes of our approach removing various percentages of compiler-QPU-combinations (in seconds).

Circuit	0%	50%	70%	90%
Gr3	84.34	51.72	42.01	31.56
RC4	108.80	63.05	51.49	39.32
RC5	109.81	65.65	54.39	42.88

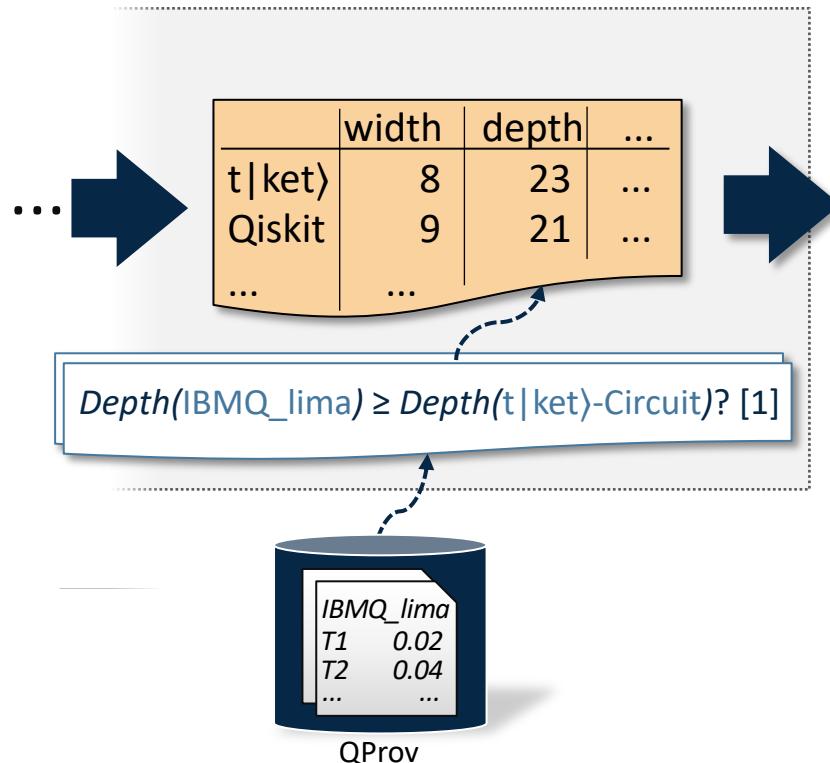


62% faster

Translation & Compilation of Quantum Circuits



Analysis of the Executability of Compilation Results

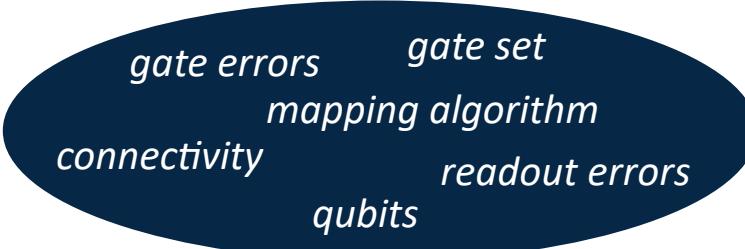


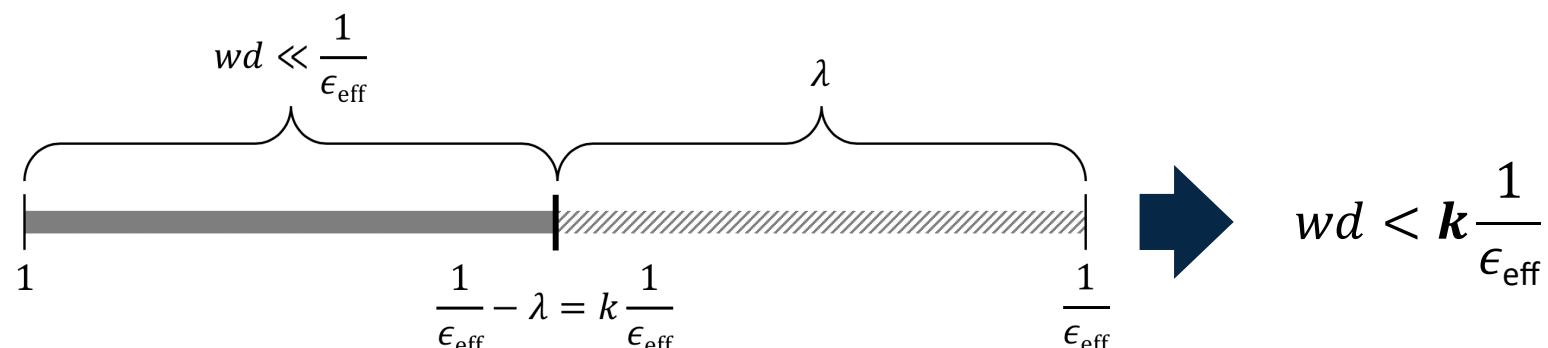
Backend Name	Provider	SDK	Width	Depth	Multi-Qubit Gate Depth	Total Number of Operations
ibmq_qasm_simulator	ibmq	pytket	3	14	6	21
ibmq_bogota	ibmq	pytket	3	27	9	45
ibmq_bogota	ibmq	qiskit	3	30	13	51
ibmq_lima	ibmq	pytket	3	27	9	45

- Salm, Marie; Barzen, Johanna; Leymann, Frank; Weder, Benjamin; Wild, Karoline: Automating the Comparison of Quantum Compilers for Quantum Circuits. In: Proceedings of the 15th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2021), 2021.
- Salm, Marie; Barzen, Johanna; Breitenbücher, Uwe; Leymann, Frank; Weder, Benjamin; Wild, Karoline: The NISQ Analyzer: Automating the Selection of Quantum Computers for Quantum Algorithms. In: Proceedings of the 14th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2020), Springer International Publishing, 2020.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Weder, Benjamin: Prioritization of Compiled Quantum Circuits for Different Quantum Computers. In: Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2022), IEEE, 2022.
- Weder, Benjamin; Barzen, Johanna; Leymann, Frank; Salm, Marie; Wild, Karoline: QProv: A provenance system for quantum computing. In: IET Quantum Communication. Vol. 2(4), Wiley Online Library, 2021.

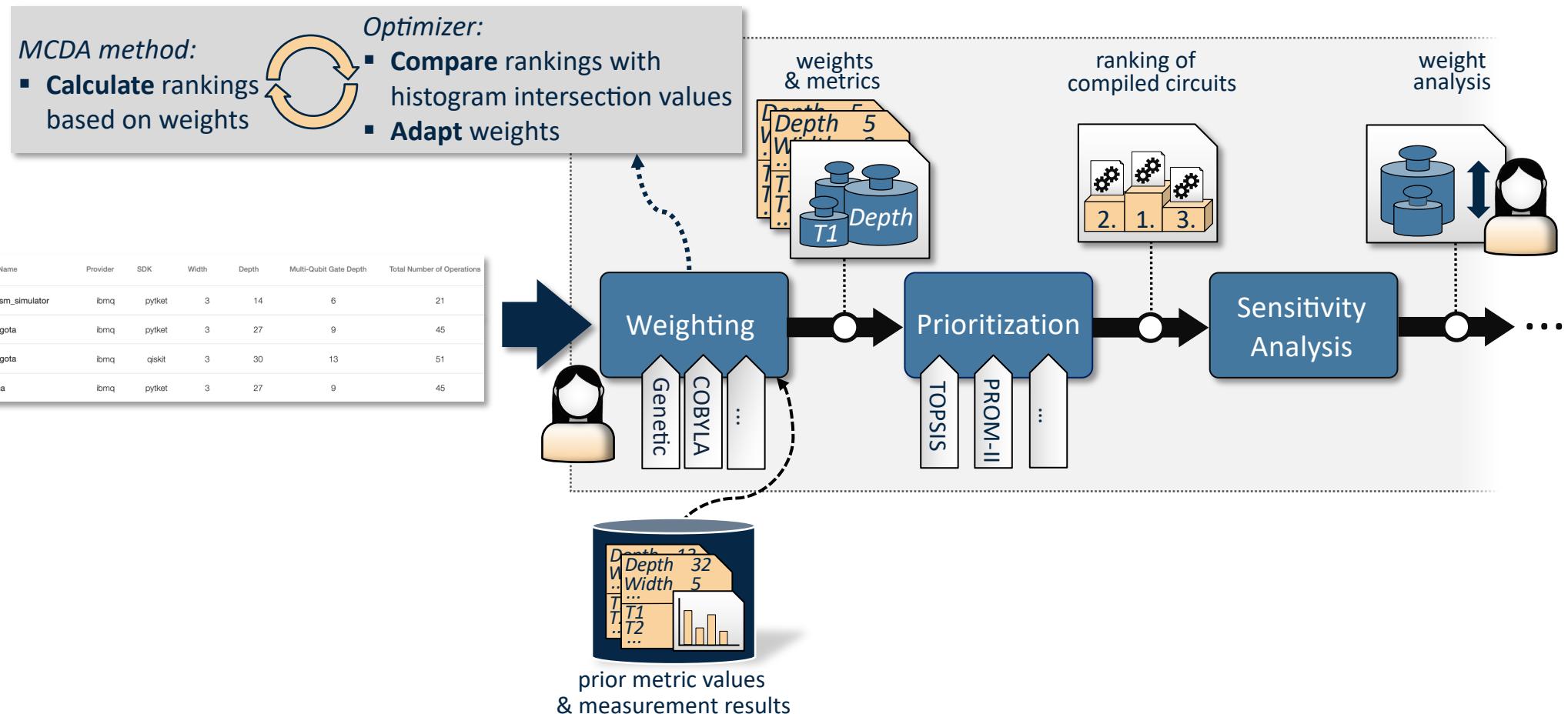
Analysis of the Executability of Compilation Results

$$wd \ll \frac{1}{\epsilon_{\text{eff}}} [2,3]$$

- w : width of circuits, d : depth of circuit, ϵ_{eff} : effective error rate
- $wd \approx \frac{1}{\epsilon_{\text{eff}}} \rightarrow$ execution probably fail [4]
- Composition of ϵ_{eff} :

 - gate errors
 - gate set
 - mapping algorithm
 - connectivity
 - readout errors
 - qubits
- Sharpening the equation from \ll to $<$:

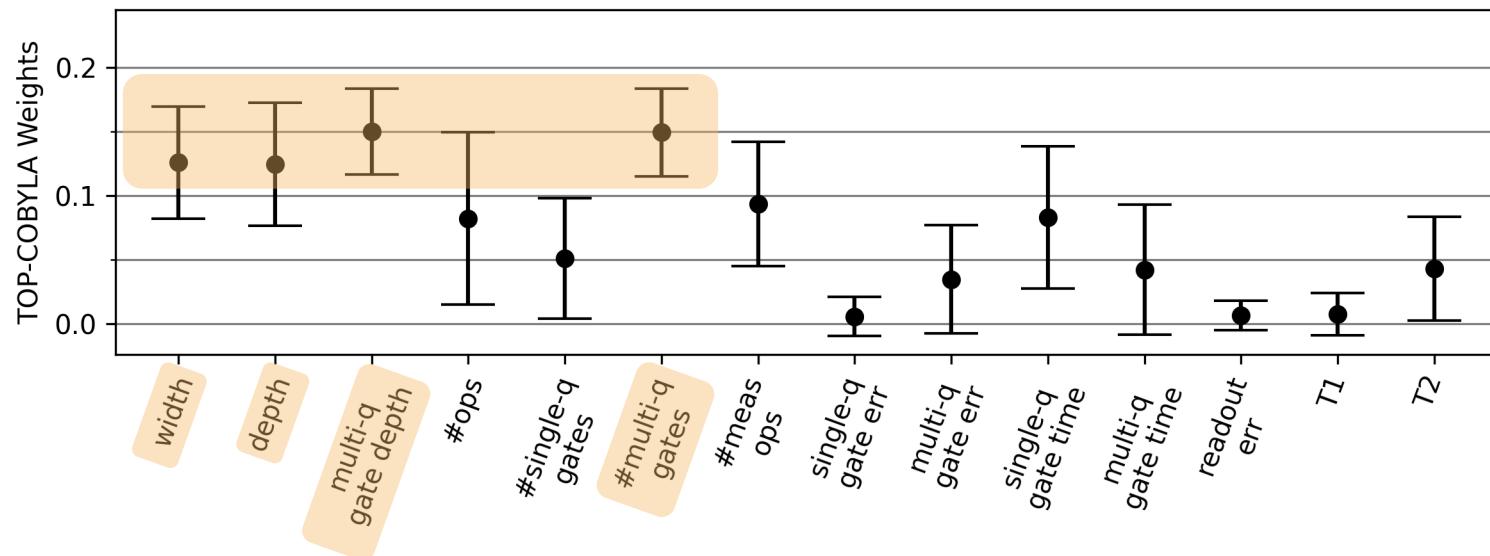


Preference-based Prioritization of Compilation Results

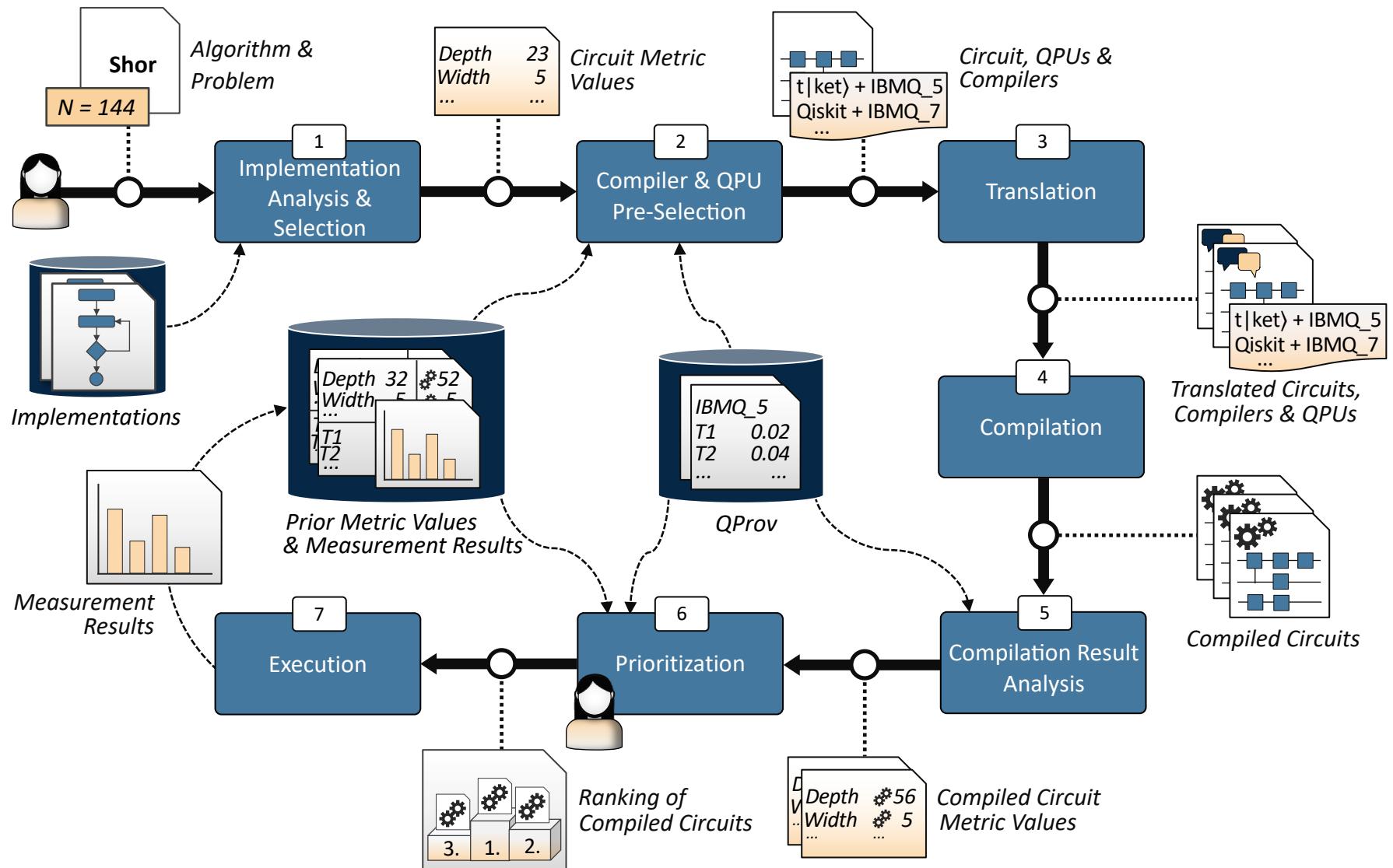


- Salm, Marie; Barzen, Johanna; Leymann, Frank; Weder, Benjamin: Prioritization of Compiled Quantum Circuits for Different Quantum Computers. In: Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2022), IEEE, 2022.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Wundrack, Philipp: Optimizing the Prioritization of Compiled Quantum Circuits by Machine Learning Approaches. In: Proceedings of the 16th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2022), Springer, 2022.

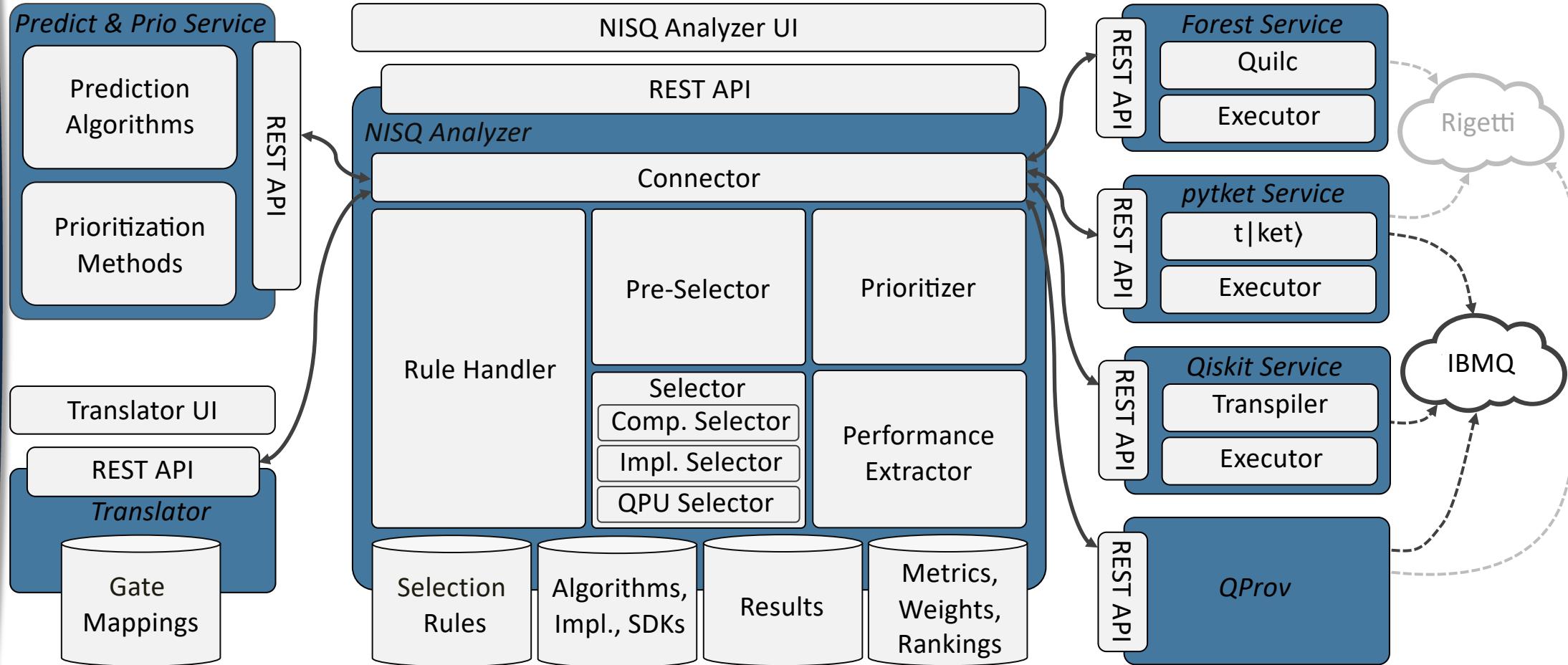
Preference-based Prioritization of Compilation Results – Case Study



Approach of the NISQ Analyzer



Architecture



How It Looks ...

Use Case 1: Analysis Based on Algorithm

The screenshot shows a web-based application for managing quantum algorithm analyses. At the top, there is a navigation bar with tabs: General, Implementations, Related algorithms, Publications, NISQ Analyzer (which is currently selected), and NISQ Results. To the right of the tabs are three buttons: a plus sign (+), Factorization (with a close button), and a help icon (?). Below the navigation bar, the main content area has a title "Analysis Jobs" and a blue "New Analysis" button. A modal window titled "Start Analysis" is open in the center. It contains three input fields: "N" (with the value "1"), "Cloud Service" (with a dropdown menu showing "None" and "AWS"), and "Your Token" (with an empty input field). At the bottom of the modal are two buttons: "Cancel" and "Ok". In the background, there is a table listing six analysis jobs. The columns are "Time" and "Readiness". Each job entry includes a "Show analysis" button. The table rows are as follows:

Time	Readiness
2022-06-02T18:40:36.224119Z	Show analysis
2022-06-02T19:05:57.263834Z	Show analysis
2022-06-02T19:10:31.049747Z	Show analysis
2022-06-02T19:45:33.816735Z	Show analysis
2022-06-02T19:47:20.595699Z	Show analysis
2022-06-02T19:54:22.10116Z	Show analysis

Compilation Results of Different Implementations

Implementation: shor-fix

Backend Name	Provider	SDK	Width	Depth	Multi-Qubit Gate Depth	Total Number of Operations	Number of Single-Qubit Gates	Number of Multi-Qubit Gates	Number of Meas
ibmq_qasm_simulator	ibmq	qiskit	5	7	4	10	2	5	
ibmq_santiago	ibmq	qiskit	5	15	3	24	15	6	
ibmq_bogota	ibmq	qiskit	5	17	8	26	12	11	
ibmq_lima	ibmq	qiskit	4	12	5	14	6	5	
ibmq_belem	ibmq	qiskit	5	12	7	17	6	8	
ibmq_quito	ibmq	qiskit	5	12	7	17	6	8	
simulator_statevector	ibmq	qiskit	5	7	4	10	2	5	
simulator_mps	ibmq	qiskit	5	6	4	9	1	5	
simulator_extended_stabilizer	ibmq	qiskit	3	7	2	9	4	2	
ibmq_manila	ibmq	qiskit	5	11	5	16	5	8	

Implementation: shor-general

Backend Name	Provider	SDK	Width	Depth	Multi-Qubit Gate Depth	Total Number of Operations	Number of Single-Qubit Gates	Number of Multi-Qubit Gates	Number of Meas
ibmq_qasm_simulator	ibmq	qiskit	5	7	4	10	2	5	
ibmq_santiago	ibmq	qiskit	4	18	6	25	16	6	
ibmq_lima	ibmq	qiskit	4	12	5	14	6	5	

Required Information

Shor / shor-general
Quantum Algorithm Quantum Implementation

+ ?

General Publications Software Platforms Selection Criteria Execution NISQ Analyzer

Required information for the NISQ Analyzer

File Location
[https://raw.githubusercontent.com/UST-QuAntiL/nisq-analyzer-content/master/example-implementations/Shor/...](https://raw.githubusercontent.com/UST-QuAntiL/nisq-analyzer-content/master/example-implementations/Shor/)

SDK
Qiskit

Language
Qiskit

Input Parameters

Name of the Parameter	Description	Datatype	Restriction
N	Integer to be factored	Integer	N > 2

Prolog Rules

Selection Rule (in PROLOG)

```
processable(N, shor-general) :- N > 2, 1 is mod(N, 2).
```

Use Case 2: Analysis Based on Implementation



The screenshot shows a web-based interface for managing quantum computing jobs. In the background, there's a list of 'QPU Analysis Jobs' with a button to 'New Analysis'. The main focus is a modal dialog titled 'Start QPU-Selection-Analysis'.

Vendor *: IBMQ

Token: (Input field)

Select preferences:

- Short Waiting Times
- Precise Execution Results

Advanced settings: (Dropdown menu)

Select Importance R: 40:60 (A slider with a blue dot at the center, indicating a 50-50 split between 'Short Waiting' and 'Precise Results').

Maximum number of compilation results: 4

Select SDKs to be used for compilation:

- pytket
- qiskit

Cancel **Ok**

Analysis Results

Randomized-4-Qubit
Algorithm
/ pattern0-3_2-1nCliffs1seed2
Implementation

+ **(?)**

General Publications Software Platforms Selection Criteria Execution **NISQ Analyzer**

Back **Prioritize**

QPU Analysis Job from 2022-11-17T16:37:39.966644+01:00

Rank	Score	Backend Name	Provider	SDK	Width	Depth	Multi-Qubit Gate Depth	Total Number of Operations	Number of Single-Qubit Gates
-	-	ibmq_manila	ibmq	pytket	4	19	2	53	45
-	-	ibm_oslo	ibmq	pytket	4	19	2	53	45
-	-	ibmq_manila	ibmq	qiskit	4	22	4	50	40
-	-	ibm_nairobi	ibmq	pytket	4	19	2	53	45
-	-	ibmq_qasm_simulator	ibmq	qiskit	4	11	4	28	18

Prioritization

Randomized-4-Qubit

Algorithm

/ pattern0-3_2-1nCliffs1seed2

Implementation

General Publications Software

Back

QPU Analysis Job from 2

Rank	Score	Backend Name
-	-	ibmq_manila
-	-	ibm_oslo
-	-	ibmq_manila
-	-	ibm_nairobi
-	-	ibmq_qasm_simulator

NISQ Analyzer

1:00

(ns)	Queue Size	Est. Histogram Intersection Value	Execution Result
0.812	120	0.81847817	Execute
0.95	65	0.8091585	Execute
0.812	120	0.8014575	Execute
0.73	211	0.79373777	Execute
	0	-	Execute

Prioritize Analysis Results

Select predefined preferences

Short Waiting Times

Precise Execution Results

Advanced settings ▾

Select Importance Ratio:

Short Waiting Precise Results

Or define individual metric weights

Cancel OK

Ranked Results

Randomized-4-Qubit
Classic Algorithm
/ pattern0-3_2-1nCliffs1seed2
Classic Implementation

+ Ⓞ

General	Publications	Software Platforms	Selection Criteria	Execution	NISQ Analyzer

[Back](#) Prioritize [Analyze Rank Sensitivity](#)

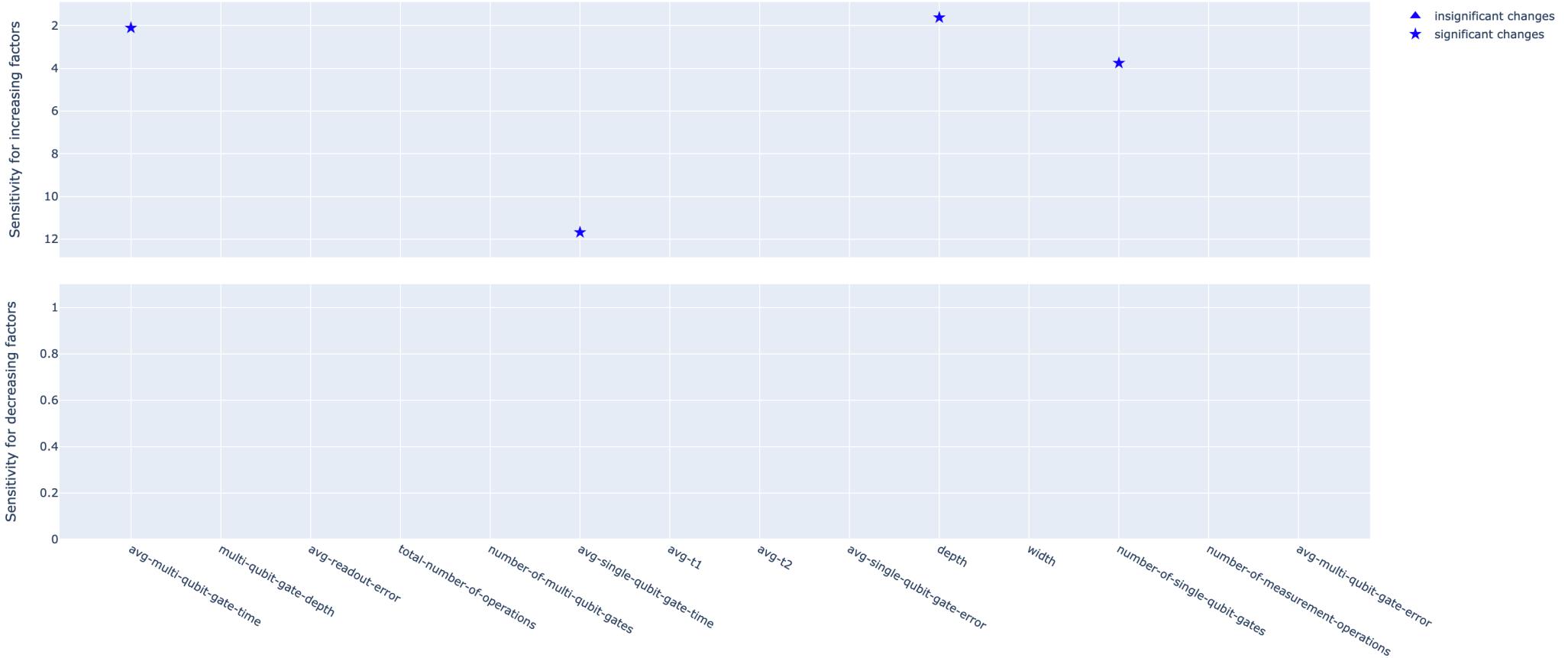
QPU Analysis Job from 2022-11-17T16:37:39.966644+01:00

Rank	Score	Backend Name	Provider	SDK	Avg. T1 (ns)	Avg. T2 (ns)	Queue Size	Est. Histogram Intersection Value	Execution Result
1	-	ibmq_oslo	ibmq	pytket	151106.8	87569.5	65	0.8091585	Execute
2	-	ibmq_nairobi	ibmq	pytket	105642.945	80908.73	211	0.79373777	Execute
3	-	ibmq_manila	ibmq	pytket	192275.75	52704.812	120	0.81847817	Execute
4	-	ibmq_qasm_simulator	ibmq	qiskit	-	-	0	-	Execute
5	-	ibmq_manila	ibmq	qiskit	192275.75	52704.812	120	0.8014575	Execute

Analysis of Ranking

Sensitivity Analysis

Each metric weight was adapted based on the previously defined parameters. A triangle of a metric represents the factor by which the original weight was adapted and for which a change of the original ranking was detected. The closer a triangle is to 1, the more sensitive is the respective metric weight.



How It Feels ...

Evaluation

- Implementation constraints via Prolog hard
- Desired features
 - export of compilation & execution results
 - error correction & mitigation
 - variational algorithms
- Positive feedback
 - comparison
 - prioritization
 - time saving
 - recognizing dimensions & differences between compilers

Conclusion & Future Work

- Automated selection & recommendation of quantum resources
 - Translation of quantum circuits
 - Compilation with different quantum compilers
 - Prediction of execution result precision
 - Plug-in based, supports extensibility
- In the future
 - Additional circuits, compilers, QPUs
 - Further metrics, e.g., monetary metrics

Thank you!

References

- [1] Sete, E.A., Zeng, W.J., Rigetti, C.T.: A functional architecture for scalable quantum computing. In: 2016 IEEE International Conference on Rebooting Computing (ICRC). pp. 1–6 (2016)
- [2] Frank Leymann and Johanna Barzen. 2020. The bitter truth about gate-based quantum algorithms in the NISQ era. *Quantum Science and Technology* 5, 4 (2020), 044007.
- [3] John Preskill. 2018. Quantum Computing in the NISQ era and beyond. *Quantum* 2 (2018), 79.
- [4] Nikolaj Moll, Panagiotis Barkoutsos, Lev S Bishop, Jerry M Chow, Andrew Cross, et al. 2018. Quantum optimization using variational algorithms on near-term quantum devices. *Quantum Science and Technology* 3, 3(2018), 030503.

Our Publications About the NISQ Analyzer & Co.

- Salm, Marie; Barzen, Johanna; Breitenbücher, Uwe; Leymann, Frank; Weder, Benjamin; Wild, Karoline: **The NISQ Analyzer: Automating the Selection of Quantum Computers for Quantum Algorithms**. In: Proceedings of the 14th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2020), Springer International Publishing, 2020.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Weder, Benjamin: **About a Criterion of Successfully Executing a Circuit in the NISQ Era: What $wd \ll 1/\varepsilon_{\text{eff}}$ Really Means**. In: Proceedings of the 1st ACM SIGSOFT International Workshop on Architectures and Paradigms for Engineering Quantum Software (APEQS 2020), ACM, 2020.
- Weder, Benjamin; Barzen, Johanna; Leymann, Frank; Salm, Marie: **Automated Quantum Hardware Selection for Quantum Workflows**. In: Electronics. Vol. 10(8), MDPI, 2021.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Weder, Benjamin; Wild, Karoline: **Automating the Comparison of Quantum Compilers for Quantum Circuits**. In: Proceedings of the 15th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2021), 2021.
- Weder, Benjamin; Barzen, Johanna; Leymann, Frank; Salm, Marie; Wild, Karoline: **QProv: A provenance system for quantum computing**. In: IET QuantumCommunication. Vol. 2(4), Wiley Online Library, 2021.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Weder, Benjamin: **Prioritization of Compiled Quantum Circuits for Different Quantum Computers**. In: Proceedings of the 2022 IEEE International Conference on Software Analysis, Evolution and Reengineering (SANER 2022), IEEE, 2022.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Wundrack, Philipp: **Optimizing the Prioritization of Compiled Quantum Circuits by Machine Learning Approaches**. In: Proceedings of the 16th Symposium and Summer School on Service-Oriented Computing (SummerSOC 2022), Springer, 2022.
- Salm, Marie; Barzen, Johanna; Leymann, Frank; Wundrack, Philipp: **How to Select Quantum Compilers and Quantum Computers Before Compilation**. In: Proceedings of the 13th International Conference on Cloud Computing and Services Science (CLOSER 2023), SciTePress, 2023 (accepted for publication).



<https://www.iaas.uni-stuttgart.de/en/institute/team/Salm/>