Application-Centric Benchmark Verification Report

Verified by

KEARNEY

Contents

Report Overview 3

Overview of TTS Benchmarks 3

Verification Methodology 3

Limitations of the Verification Methodology and Recommendations for Future Work 4

  Time-to-Solution Estimation 4

  Benchmark Interpretability 5

QFT Benchmark Verification 6

  Algorithm Overview and Definitions 6

  Verification Results 6

Hidden Shift Benchmark Verification 7

  Algorithm Overview and Definitions 7

  Verification Results 8

LR-QAOA Benchmark Verification 9

  Algorithm Overview and Definitions 9

  Verification Results 9

VQE Benchmark Verification 10

  Algorithm Overview and Definitions 10

  Verification Results 11

Conclusion 11

Legal Disclaimer 12

Report Overview

This report provides a verification of application-centric benchmark performance of IonQ Forte and competitor systems, particularly focused on Time-To-Solution (TTS) for key algorithms.

Overview of TTS Benchmarks

Inspired by the machine learning benchmarking suite MLPerf, IonQ has developed a set of application-centric benchmarks to assess real-world-representative performance for significant use cases. The key metric for evaluating performance for these benchmarks is Time-to-solution (TTS), defined either as:

(a) how quickly a quantum computer can reach a solution for a defined quality target.

(b) the execution time for a quantum algorithm that should theoretically find a solution for a defined quality target

For the following algorithms, time-to-solution based on definition (a) was evaluated for a fixed quality threshold defined for each benchmark, both for IonQ Forte and competitor systems:

- *Quantum Fourier Transform (QFT)*
- *Hidden Shift Benchmark Problem (HSBP)*
- *Linear-Ramp Quantum Approximate Optimization Algorithm (LR-QAOA)*

In addition, the *Variational Quantum Eigensolver (VQE)* algorithm performance was evaluated based on the definition (b) across IonQ and competitor systems.

Verification Methodology

This verification report was prepared with the assistance of IonQ's Applications team. IonQ provided access to all datasets and analysis software organized into a shared repository on Github and Google Drive folders.

We adopt a structured verification framework encompassing code reviews, reproduction of results from the original sample distributions, and validation runs on IonQ hardware.

1) *Review of benchmark quality scoring code* to ensure code correctly implements the benchmark score definitions as defined by IonQ
2) *Review of TTS estimation code* to ensure that TTS is calculated consistently with how it is formally defined by IonQ
3) *Reproduce competitor benchmarks from original sample distributions* to ensure reproducibility of the benchmark quality and TTS calculations and to ensure consistency and unbiasedness of the underlying data.

4) *Reproduce IonQ benchmarks using the original sample distributions* to ensure reproducibility of TTS calculations on the underlying data
5) *Validate IonQ sample distributions and benchmark measurements on IonQ hardware* to ensure reproducibility and end-to-end validation of the TTS theory, code implementation, and hardware execution.

Limitations of the Verification Methodology and Recommendations for Future Work

We note several limitations of this verification. First, for software validation, we assume that the implemented quantum circuits faithfully reflect their corresponding theoretical specifications and do not undertake independent verification or debugging of the quantum code. Our evaluation instead emphasizes the consistent and unbiased execution and measurement of the quantum circuits.

Time-to-Solution Estimation

For the three TTS benchmarks (QFT, HSBP, LR-QAOA), TTS is estimated by proxy through a fixed time-per-shot (t_{shot}) multiplied by estimated number of shots (N_{shots}). The estimated number of shots (N_{shots}), is calculated as $N_{shots} = \frac{\ln 0.01}{\ln(1-p_s)}$ with p_s as the probability of obtaining at least one solution at or above a given quality threshold at 99% confidence. The time-per-shot (t_{shot}) was estimated from runs with a large number of shots on both IonQ and competitor systems – while this does not change the overall conclusions on IonQ performance, particularly at high quality thresholds, it does potentially create deviations from true TTS values. These potential deviations are due to: a) understatement of initialization times (for compile, transpile, prepare, and load times), which could cause the actual TTS measurements (particularly on shorter runs) to differ from what is reported, and b) potential discrepancies in how the total execution time was measured on IonQ hardware vs competitor systems. Moreover, the published material reports TTS and quality scores without accompanying error ranges, which deviates from established best practices in reporting. Best practice generally calls for including measures of uncertainty alongside point estimates; their omission limits the reader’s ability to assess variability, statistical significance, and the robustness of the results, thereby reducing transparency and interpretability.

There are substantial challenges in establishing a directly comparable, like-for-like assessment of time-to-solution across different providers. For example, one provider might divide a run into separate batches and average them, whereas another provider might follow a different execution process. This makes a clear view and comparison of initialization times across providers difficult to assess. Therefore, while there are limitations of estimating TTS from a fixed time-per-shot, it is an acceptable method given the difficulties of cross-provider comparisons.

There is potential to address these challenges through industry-wide standardization of execution time reporting requirements for benchmarks. A neutral third party consortium, similar to MLCommons for the MLPerf benchmarks, could ensure that time-to-solution and other quantum benchmarks are reported according to standard, industry-wide rules. This will not only ensure fair comparison of competing systems, but will help accelerate quantum computing development. A third-party consortium for benchmarking can also develop and test a wider set of benchmarks, and provide a more comprehensive view of provider performance.

Benchmark Interpretability

The scope of this verification is limited to assessing whether the procedures employed by the IonQ team in measuring the application-centric benchmarks were conducted in accordance with the defined benchmark rules - the verification does not evaluate the appropriateness of these benchmarks for assessing quantum computer performance more broadly. We note a couple key considerations for future work regarding the definition of such benchmarks to improve interpretability.

First, there are potential opportunities to better facilitate comparisons across benchmarks through normalization of benchmark values versus a random baseline. For example, in assessing quality scores – a randomly noisy device will produce a benchmark score approaching 1.0 for QFT, but will produce a score approaching 0.0 for Hidden Shift due to the formulation of the algorithms. This poses issues when trying to compare IonQ performance across multiple benchmarks – normalization of the scoring across benchmarks (either against a random baseline or other metric) could improve interpretability and provide a clearer view of overall IonQ performance.

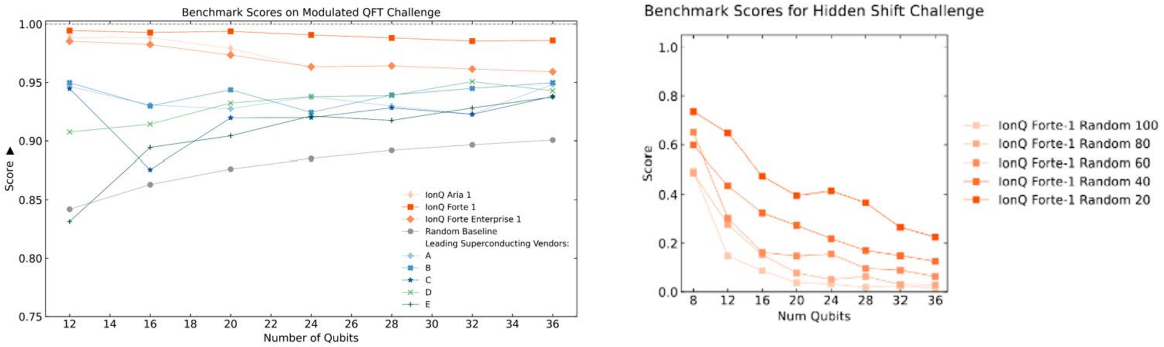

Figure 1 - Quality score comparison across providers for QFT and Hidden Shift

In addition, the root causes behind the varying performance of IonQ across benchmarks are only lightly touched on in existing publications and present an opportunity for clearer communication and understanding of performance differences in the future. In particular, while error mitigation was applied during the benchmarking, the exact specifications and techniques applied are not specified. Spot checks conducted on IonQ hardware, as well as runs on competitor systems, were performed using each provider's standard error correction protocols. As such, the reported performance estimates are intended to be representative of what a typical customer would experience under normal operating conditions. Nonetheless, there is an opportunity for future benchmarking to more clearly specify the exact techniques to ensure consistency.

QFT Benchmark Verification

Algorithm Overview and Definitions

Performance for the Quantum Fourier Transform was benchmarked through applying a reference cosine wave and recovering the encoded frequency of the wave. This can be formulated as:

$$\sum_k \cos\left(\frac{2\pi k}{N} s\right) |k\rangle \rightarrow \frac{1}{\sqrt{2}}(|s\rangle + |-s\rangle)$$

Where the time-domain cosine amplitudes are decomposed into the frequencies $+s$ and $-s$. For this particular benchmark test, the frequency s is set to $s = 2^{n/2} - 1$, where n is the number of qubits. The TTS benchmarking was evaluated on instances of 36 qubits.

The quality score is the Hamming distance between the output and nearest target (either the positive or negative frequency s). The TTS benchmark for a given Hamming distance is estimated as $TTS = N_{shots} \cdot t_{shot}$ where N_{shots} is the estimated number of shots to provide at least one solution of that distance (or shorter) with 99% confidence, and t_{shot} is the average time per shot. t_{shot} is calculated as 0.337406 s for IonQ Forte calculated from 1,000 shots, and 472 µs for the leading competitor, calculated from 10 runs of 100,000 shots each.

Verification Results

Check	Results	Comments
Review of benchmark quality scoring code	Pass	Code implementation adheres to published definitions of Hamming distance
Review of TTS estimation code	Pass	Code implementation adheres to definition of TTS with estimated # shots to achieve an answer with given Hamming distance with 99% confidence
Reproduce competitor benchmarks and check original sample distributions	Pass	Post-processing code reproduces competitor benchmarks exactly from original distributions. Competitor sample distributions consistent with expected behavior

Reproduce IonQ benchmarks from original sample distributions	Pass	Post-processing code reproduces IonQ benchmarks exactly from original distributions
Validate IonQ sample distributions and benchmark measurements on IonQ hardware	Pass	Tested with 50 shots (total execution time: 48.278 s) on IonQ Forte – results consistent with original sample distributions

The final check serves to validate IonQ sample distributions through runs on IonQ hardware. With 50 shots executed in 48.278 s, 4 of them achieve a Hamming distance of 0, consistent with the original sample distribution (34 shots out of 1,000). We note that the time-per-shot from our test run (0.96s) is larger than that used by IonQ, which is consistent with the hypothesis that jobs with fewer shots will be more impacted by initialization time.

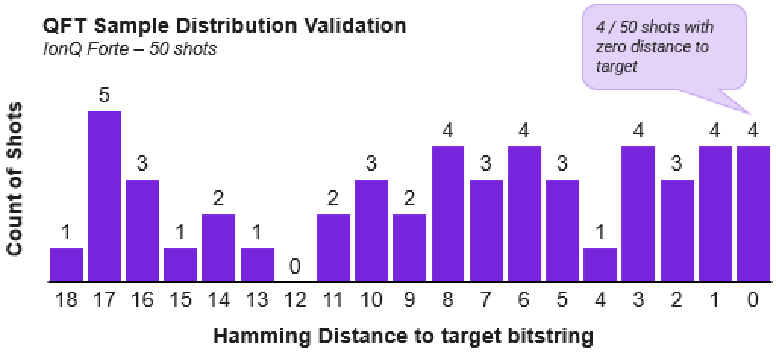

Figure 2 - Cosine QFT distribution check on IonQ hardware

Hidden Shift Benchmark Verification

Algorithm Overview and Definitions

The Hidden Shift Benchmark Problem is used to evaluate a quantum computer's ability to identify a hidden translation (s) of function *f* to another function *g*: $g(x) = f_\pi(x \oplus s)$. First, a bitwise shift s is applies to the input bitstring x. Then, the function f, in this case, a Maiorana–McFarland (MM) bent function, permutes half the bits of the input bitstring via permutation defined by π, then applies the bitwise inner-product with the other half. For the TTS benchmarks, a randomly generated permutation π was applied. The TTS benchmarking was evaluated on instances of 36 qubits.

The quality benchmark score is defined as the Hamming distance between the estimated shift bitstring and the actual (s). The TTS benchmark for a given Hamming distance is estimated as $TTS = N_{shots} \cdot t_{shot}$ where N_{shots} is the estimated number of shots to provide at least one solution of that distance (or shorter) with 99% confidence, and t_{shot} is the average time per shot. t_{shot} is calculated as 0.396689s for IonQ Forte calculated from 900 shots, and 337μs for the leading competitor, calculated from 1 run of 900,000 shots each.

Verification Results

Check	Results	Comments
Review of benchmark quality scoring code	Pass	Code implementation adheres to published definitions of Hamming distance
Review of TTS estimation code	Pass	Code implementation adheres to definition of TTS with estimated # shots to achieve an answer with given Hamming distance with 99% confidence
Reproduce competitor benchmarks and check original sample distributions	Pass	Post-processing code reproduces competitor benchmarks exactly from original distributions. Competitor sample distributions consistent with expected behavior
Reproduce IonQ benchmarks from original sample distributions	Pass	Post-processing code reproduces IonQ benchmarks exactly from original distributions
Validate IonQ sample distributions and benchmark measurements on IonQ hardware	Pass	Tested with 450 shots (total execution time: 257.17 s) on IonQ Forte – results consistent with original sample distributions

The final check serves to validate IonQ sample distributions through runs on IonQ hardware. With 9 runs, each for a different instance of the algorithm, of 50 shots each, executed in 257.17 s, 6 shots (out of 450) achieve a Hamming distance of 0, consistent with the original sample distribution (25 shots out of 900).

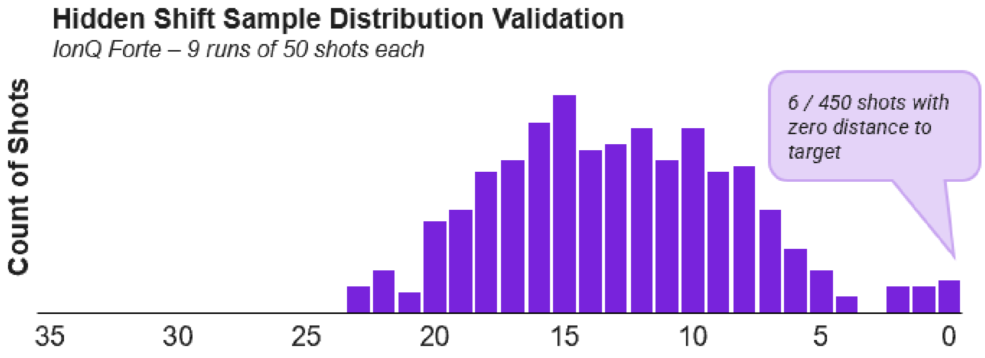

Figure 3 - Hidden Shift distribution check on IonQ hardware

LR-QAOA Benchmark Verification

Algorithm Overview and Definitions

The Linear-Ramp Quantum Approximate Optimization Algorithm (LR-QAOA) is a variation of the more standard QAOA algorithm and can be used to solve a wide range of combinatorial optimization problems.

For this benchmark, performance is measured against the MaxCut optimization problem, in this case, an LR-QAOA with 9 layers, evaluated on a MaxCut problem on a 4-regular graph with 36 qubits.

For this LR-QAOA, the quality score is defined as the approximation ratio: $AR = \frac{C(\gamma,\beta)}{C_{opt}}$ where C is the cost function, γ, β are parameters for the LR-QAOA, and C_{opt} is the exact optimal cost.

For this LR-QAOA, the TTS benchmark for a given approximation ratio target (AR_{min})is estimated as $TTS = N_{shots} \cdot t_{shot}$ where N_{shots} is the estimated number of shots to provide at least one solution with AR > AR_{min} at 99% confidence. Specifically, this is calculated as $N_{shots} = \frac{\ln 0.01}{\ln(1-p_s)}$ with p_s as the probability of obtaining at least one solution with AR > AR_{min}, estimated from the output sample distributions. The time-per-shot t_{shot} is the estimated as 0.89 s for IonQ Forte (calculated from 5,000 shots), and 445 μs for the leading competitor (calculated from 10 runs of 100,000 shots each).

Verification Results

The following table summarizes the checks performed across the verification framework. Each check is graded as Pass, Interpretation-Dependent, or Fail.

Check	Results	Comments
Review of benchmark quality scoring code	Pass	Code implementation adheres to published definition of Approximation Ratio
Review of TTS estimation code	Pass	Code implementation adheres to definition of TTS with estimated # shots to achieve an answer with AR > AR_{min} with 99% confidence
Reproduce competitor benchmarks and check original sample distributions	Pass	Post-processing code reproduces competitor benchmarks exactly from original distributions. Competitor sample distributions consistent with expected behavior

Reproduce IonQ benchmarks from the original sample distributions	Pass	Post-processing code reproduces IonQ benchmarks exactly from original distributions
Validate IonQ sample distributions and benchmark measurements on IonQ hardware	Pass	Tested with 50 shots (total execution time: 76.876 s) on IonQ Forte – results consistent with original sample distributions

The final check serves to validate IonQ sample distributions through runs on IonQ hardware. With 50 shots executed in 76.876 s, 5 of them achieve an AR greater than 90%, consistent with the original sample distribution (561 shots out of 5,000).

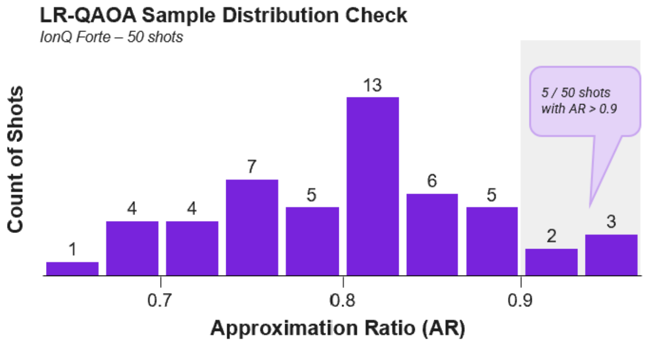

Figure 4 - LR-QAOA distribution check on IonQ hardware

VQE Benchmark Verification

Algorithm Overview and Definitions

The Variational Quantum Eigensolver (VQE) is an algorithm primarily used in quantum chemistry, to calculate the ground state energy of a molecule, with applications in drug discovery and materials science. This particular benchmark calculates the ground state energy for chains of hydrogen atoms. The benchmark quality score is measured as the absolute difference in estimated energy ($E_{measured}$) vs the exact value (E_{DOCI})

$$Error = |E_{measured} - E_{DOCI}|$$

The benchmark was tested on instances comprised of 2 to 18 hydrogen atoms.

Verification Results

Check	Results	Comments
Review of benchmark quality scoring code	Pass	Code implementation adheres to published definition of energy error
Reproduce competitor benchmarks (execution time) and spot check original sample distributions	Pass	Post-processing code reproduces competitor benchmark values
Spot check execution time by running on IonQ hardware	Pass	Execution time (38 minutes 956 msec) consistent with reported results

The spot check (10 qubits, 2 angstrom bond length) on IonQ hardware executed in 38 min, 956 msec, and is consistent with IonQ-reported results showing no advantage in time-to-solution for VQE for IonQ hardware vs superconducting providers.

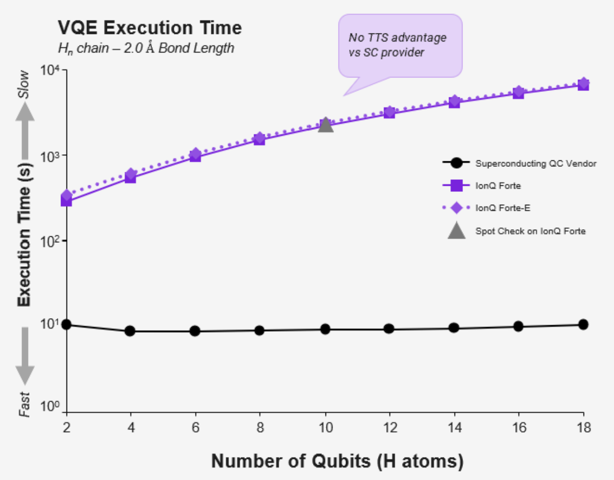

Figure 5 - VQE spot check on IonQ hardware

Conclusion

Based on the verification activities, including code reviews, post-processing validation runs and validation on IonQ hardware, it is concluded that the procedures employed by IonQ in measuring these application-centric benchmarks adhere to the defined rules and the collected data is consistent under basic validation. This verification confirms procedural compliance and data integrity within the defined scope, with results time-bound to the system's calibration state and software revision at the moment of execution. We also affirm the overall estimation methodology for time-to-solution, while acknowledging its limitations given the data available across different providers. We propose potential improvements to improve interpretability, as well as note the potential for the formation of a consortium in the same spirit as MLCommons for the purposes of standardizing and reviewing benchmarks for the quantum computing industry.

This report was prepared by A.T. Kearney, Inc ("Kearney") advised by Cascade Quantum, Inc. For further questions, please reach out to Brent Smolinski (brent.smolinski@kearney.com)

Legal Disclaimer

This Verification Report (the "Report") was prepared at the request of IonQ. The Report is provided subject to the terms of the engagement agreement between Kearney and IonQ and is intended solely for informational purposes. Kearney does not accept any liability or responsibility to any third party who may receive or access this Report, including but not limited to IonQ's shareholders, counterparties, regulators, or government agencies.

This Report may contain forward-looking statements, projections, estimates, or other predictive information. Such information is inherently speculative and subject to numerous risks, uncertainties, and assumptions, which are beyond Kearney's control. Actual results may differ materially from those expressed or implied in this Report. Kearney makes no representation or warranty, express or implied, as to the accuracy, completeness, or reliability of any information contained herein, including any financial data, forecasts, or conclusions.

Kearney has relied upon information provided by IonQ and publicly available sources. Kearney does not guarantee the accuracy of such information and does not provide accounting, tax, legal, or investment advice to IonQ. The Report does not constitute a recommendation or endorsement of any strategy, investment, or transaction.

This Report is provided on a non-reliance basis. Neither this Report nor any of its contents may be quoted, referred to, excerpted, reproduced, disseminated, or disclosed in whole or in part, in any public filing (including filings with the U.S. Securities and Exchange Commission), investor presentation, offering memorandum, registration statement, grant application, or other document or communication, without the prior written consent of Kearney. Any such use must include this disclaimer in its entirety.

Recipients of this Report are advised to conduct their own independent analysis and consult with their own legal, financial, and other professional advisors before making any decisions based on the contents of this Report. Kearney expressly disclaims any and all liability for any loss or damage arising from reliance on or use of this Report or its contents by any party other than IonQ.

Kearney undertakes no obligation to update or revise the Report to reflect events, circumstances, or changes that occur after the date of issuance.