QUANTINUUM

Quantinuum Systems User Guide

Version 1.40 July 5, 2023

TABLE OF CONTENTS

INTRODUCTION

The **Quantinuum H-Series** hardware is the best-in-class trapped ion quantum computers. We are delighted that you've decided to explore its use for your development of quantum algorithms and solutions. This user guide covers all that you need to get up and running.

FEATURES

The complete set of the Quantinuum specifications and operations can be found in the *Quantinuum System Model H2 Product Data Sheet, Quantinuum System Model H2 Emulator Product Data Sheet, Quantinuum System Model H1 Product Data Sheet* and the *Quantinuum System Model H1 Emulator Product Data Sheet*.

ACCESS

Three systems are available to subscription users: the System Model H-Series quantum computers, System Model H-Series emulators, and device-specific syntax checkers. Users can find information about Quantinuum's systems, view submitted jobs, look up machine availability, view remaining credits, and update job notification preferences on the Quantinuum User Portal. All systems are accessible via the Quantinuum API. Directions for getting started with the API are found under the *Examples* tab on the portal.

Depending on the subscription a user has access to, the user may or may not have access to all the systems listed in the following sections. To find which systems a user has access to, navigate to the *Account* tab on the user portal.

Quantum Computers

Three trapped ion quantum computers are available to users utilizing the System Model H2 and System Model H1 generation of quantum computers, powered by Honeywell.

- H2-1 (machine target: `H2-1`)
- H1-1 (machine target: `H1-1`)
- H1-2 (machine target: `H1-2`)

Both System Model H2 and H1 are not locked into a single configuration, but rather allow for all-to-all-connectivity of qubits in three or more parallel gate zones. The System Model H1 architecture consists of a linear trap, while the System Model H2 architecture consists of a loop architecture with two linear traps. The base trap architecture for both hardware generations is known as a Quantum Charge-Coupled Device (QCCD), and more information can be found here: Demonstration of the QCCD trapped-ion quantum computer architecture, A Race Track Trapped-Ion Quantum Processor.

Individual machines that use the System Model H2 generation of ion traps are designated by the H2-X nomenclature while machines using the System Model H1 generation of ion traps are designated by the H1-X nomenclature. System Model H1-1 and System Model H1-2 are two separate quantum computing machines both using the same ion trap design. Users should expect that both systems have the same functionality, but at times may have a different number of parallel gate zones, maximum available qubits and specific performance

metrics. At all times all quantum computers will meet the minimum requirements in the *System Model H2 Product Data Sheet* and *System Model H1 Product Data Sheet*.

Emulators

To support quantum algorithm development and design, emulators are available that model each machine's specific ion transport and error rates.

- H2-1 Emulator (machine target: `H2-1E`)
- H1-1 Emulator (machine target: `H1-1E`)
- H1-2 Emulator (machine target: `H1-2E`)

The emulators operate on a physical model as well as include a detailed error model of each H-Series machine. In addition, options are provided to the user to experiment with the noise parameters of the emulator. See the *Quantinuum System Model H2 Emulator Product Data Sheet* and *Quantinuum System Model H1 Emulator Product Data Sheet* for complete details on the performance and specification of the emulators.

Benefits of using an emulator during development include: debugging code before running on quantum hardware, optimizing code in the presence of noise, exploring new quantum algorithms, and experimenting with system features such as mid-circuit measurement.

Users should expect a high degree of fidelity between the hardware and emulators, but no noise model can perfectly characterize a complex system. If you see a significant variance between the emulator and circuits run on the hardware, please contact QCsupport@quantinuum.com.

Emulator credits are tracked separately from hardware credits. The monthly allocation of each is subscription dependent.

Syntax Checkers

The syntax checkers are provided to check program syntax and are specific to each device. Before submitting quantum circuits to the Quantinuum quantum computers or emulator, use of a syntax checker is *highly* recommended. Jobs submitted to the Quantinuum hardware are in OpenQASM format. The syntax checker enables checking of the code to ensure it will run before being submitted to the quantum computers or emulator. Access to the syntax checkers for the quantum computers a user a user has access to is included.

- H2-1 Syntax Checker (machine target: `H2-1SC`)
- H1-1 Syntax Checker (machine target: `H1-1SC`)
- H1-2 Syntax Checker (machine target: `H1-2SC`)

If the code compiles, the syntax checker will return a "completed" status, the cost of the circuit in H-System Quantum Credits (HQCs), and results of all "`00`". If the code does not compile, the validator will return a "failed" status and give the error returned.

WORKFLOW

Signing In

Once a user's email address has been set up with an account on Quantinuum Systems, an invitation email will be sent with a unique signup link to the Quantinuum User Portal (https://um.qapi.quantinuum.com/). To complete registration, users will need to choose how they would like to sign in: either by creating a Quantinuum account or by using third party credentials. Currently Microsoft accounts are the only third party accounts supported. Once registration is completed, users will need to log into the user portal to view and accept our terms and conditions before they can fully access Quantinuum systems.

When using the python API wrapper provided under the *Examples* tab on the Quantinuum User Portal, you will be prompted to enter your username and password 1/month. Other APIs, such as TKET may have different policies for how often your username and password need to be entered.

Queue

When jobs are submitted, they are placed into a queue where they are run based on a fair queuing system. Jobs will wait in the queue until they run on the target device. A fair queuing process is used to ensure each organization's queue is equally represented for machine access. The jobs submitted by users in the same organization are executed in the submission order. If users submit a job to a specific machine that is not available, the jobs will remain at the top of an organization's queue until that machine is available.

Machines do not need to be online when submitting jobs. Users are encouraged to submit jobs at their convenience. The H-Series quantum computers are periodically taken down for upgrades. If a job is submitted while the machine is in an upgrade cycle, it will remain in the queue and run when the machine is back online. Users are able and encouraged to submit jobs to the queue.

Compilation

Circuits submitted to Quantinuum H-Series quantum computers and emulators are run through TKET compilation passes for H-Series hardware. This enables circuits to be automatically optimized for H-Series systems and run more efficiently.

More information on compilation passes applied can be found in the pytket-quantinuum documentation, specifically the Default Compilation section. The default compilation setting is optimization level 2. If users desire to use a different optimization level, to turn all optimizations off, or to explore what optimization passes in TKET will do before submitting, examples of how to do this can be found in the *Circuit Submissions.ipynb* and *Circuit Submissions via pytket.ipynb* Jupyter notebooks on the user portal. The *Quantinuum Application Programming Interface (API) Specification* contains the listing of all related options in the Job Submissions API.

Job Batching

Quantinuum systems support the ability to run job batches. The batch feature gives users the ability to create "ad-hoc" reservations. Circuits submitted together in a batch will run at one time. The benefit to users is that once a batch hits the front of the queue, jobs in a batch will run uninterrupted until they are completed.

Once a batch is submitted, jobs can continue to be added to the batch, ending either when the user signifies the end of a batch or after 1 minute of inactivity.

Batches cannot exceed the maximum limit of 2,000 HQCs total. If the total HQCs for jobs in a batch hit this limit or a smaller limit set by the user, those jobs will not be cancelled. Instead, they will continue to run as regular jobs in the queue instead of as a batch.

Currently only the quantum computer and emulator targets support the batching feature. Batching is not supported on the syntax checkers.

Availability

The syntax checkers and emulators are nominally available 24 hours/day to subscribers via queued access. The syntax checkers are free to use while the emulators and quantum computers require HQCs.

The System Model H2 and H1 devices are available throughout a calendar month. The calendar of system availability is maintained on the user portal. If needed, users can check a device's status by first consulting the calendar on the user portal. This provides day-by-day scheduling of machine availability. Calendars are frequently updated to reflect best available information. Users can also check availability by using the Machine State API Group.

To ensure the highest performance of our systems, we periodically schedule and perform upgrades on the machines and their components. The machines are taken offline to perform the upgrades and conduct verification and validation tests to ensure consistent performance. The performance upgrades are then made available to our users. Typical improvements include upgrades to compute speed, number of available qubits, noise reduction, and overall system reliability.

Additional information is found in the *Quantinuum Application Programming Interface ("API") Specification* document found on the user portal.

Tracking Usage

Running a job on the System Model H1 family of hardware requires H-System Quantum Credits (HQCs). When a circuit is submitted (either to a syntax checker, emulator or quantum computer) the cost in HQCs is returned with the results. Submitting to a syntax checker does not cost HQCs, but the cost of submitting the job on a quantum computer will be returned with the result for planning purposes.

The same equation is used for hardware and emulators, but the HQCs are allocated and tracked separately. You can check your credit balance in the user portal.

A H-System Quantum Credit (HQC) is defined as:

$$HQC = 5 + \frac{N_{1q} + 10\,N_{2q} + 5\,N_m}{5000} C$$

where N_{1q} is the number of single-qubit gates, N_{2q} is the number of native two-qubit gates, N_m is the number of state preparation and measurement operations in a circuit, including the initial implicit state preparation and any intermediate and final measurements and state resets, and C is the shot count.

The Metering API group assists with billing and provides end users historical information about the jobs they have executed within a specific time frame. The Metering API is used to retrieve information about the jobs run and the number of HQCs consumed within a specific time period. Metering requests can use date range, last *X* number of days or last *Y* number of jobs.

For additional information, please refer to the *Quantinuum Application Programming Interface ("API") Specification* document that is located on the user portal.

Job Status

The *Quantinuum Application Programming Interface ("API") Specification* document that is located on the user portal contains a detailed explanation of how users can monitor job status using the Job Status API. In addition to this API, users may set notification preferences via the user portal to allow email or text notifications related to job status.

Analyzing Results

Results are given in little-endian format. As an example, a 3-bit classical register `creg` with value `'abc'` has `creg[0]=c, creg[1]=b`, and `creg[2]=a.`

If you have questions about results with the Quantinuum Systems, we may be able to provide suggestions on what specifically is happening with the hardware and ask that you contact us directly at QCsupport@quantinuum.com.

Data Retention

Data from jobs submitted are kept on the user portal for 40 days. The data retention window is the same across all targets: quantum computers, emulators, and syntax checkers. To prevent unintentional loss of data, the recommended best practice is to retrieve and save job data as soon as jobs are completed.

Upgrades

Quantinuum will issue Product Change Notifications (PCNs) to customers coincidental to when new upgrades will be made commercially available or prior to any significant system changes. PCNs will initially be emailed to active administrators and will be stored on the user portal for easy reference.

SUPPORT

Quantinuum values a deep relationship with users and welcomes any questions or feedback. For initial inquiries please use the email address QCsupport@quantinuum.com. Additional personal email addresses may be made available for rapid Q&A or for more detailed discussion.