3 IBM Q Experience: уникальная платформа для квантовых вычислений в облаке

В этой главе мы рассмотрим квантовые вычисления в облаке с помощью IBM Q Experience — первой платформы такого типа. Глава начинается с обзора Composer — веб-консоли, используемой для визуального создания схем, запуска экспериментов, исследования аппаратных устройств и многого другого. Далее вы узнаете, как создать свой первый эксперимент и запустить его на симуляторе или реальном квантовом устройстве. IBM Q Experience обладает мощным REST API для управления жизненным циклом эксперимента, и в главе будет показано, как это делать, с подробным описанием конечных точек и параметров запроса. Глава заканчивается примером практической реализации официальной библиотеки Python (названной IBMQuantumExperience) для Node JS. Эта пользовательская библиотека Node JS позволит проверить ваши навыки работы с асинхронным JavaScript и REST API. Приступим!

IBM, безусловно, занимает первое место в гонке квантовых вычислений в облаке. Они придумали действительно классную платформу для удаленного запуска экспериментов под названием Q Experience. Мне кажется или в названиях этих инструментов действительно много аналогий с теорией музыки? Проверим: визуальный редактор, используемый для создания квантовых схем, называется *Composer* («Композитор»). Не странно ли? Квантовые схемы, построенные с помощью редактора, называются *партитурами* (как в нотах), не говоря уже о том, что

визуально окно редактора очень похоже на страницу нотной тетради. Я давно играю на классической гитаре и, только взглянув на Composer, сразу вспомнил гитарную партитуру (с вентилями, напоминающими ноты). Все еще думаете, что я сумасшедший? Платформа называется Q Experience, а вы когда-нибудь слышали о Experience Джимми Хендрикса? Возможно, Composer — это сборник партитур, в котором вы создадите великолепный шедевр, чтобы все мы смогли им наслаждаться. Квантовые вычисления действительно способны изменить текущее положение дел.

Первое знакомство с IBM Q Experience

Q Experience — это платформа IBM для квантовых вычислений в облаке, и она действительно крутая. Давайте посмотрим (все материалы публикуются с разрешения © International Business Machines Corporation).

1. Создайте аккаунт на https://quantumexperience.ng.bluemix.net/qx/experience. Вам понадобится указать адрес электронной почте. Затем дождитесь письма-подтверждения и продолжите регистрацию.
2. Войдите в веб-консоль и перейдите на вкладку Composer вверху (рис. 3.1).

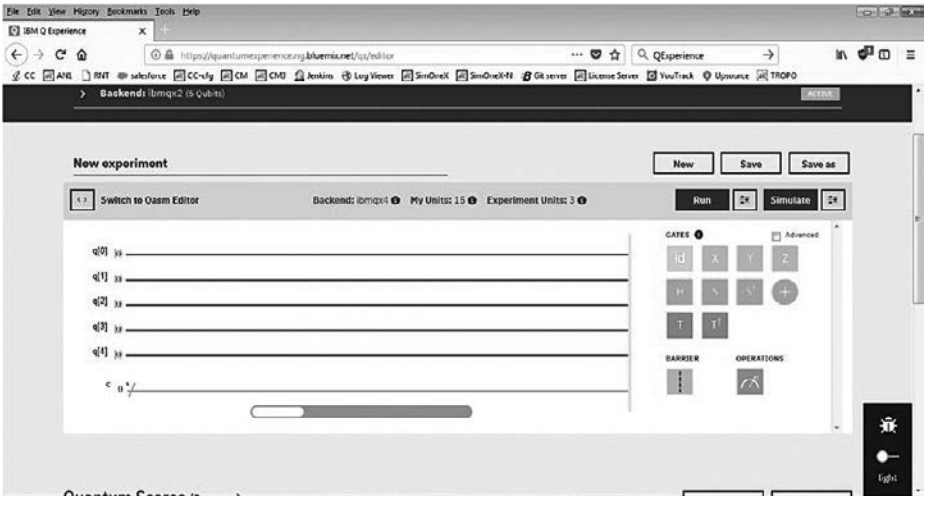

Рис. 3.1. Главное окно IBM Q Experience

Квантовый Composer

Composer — это визуальный инструмент для создания квантовых схем, или партитур. Вверху показана гистограмма эксперимента с доступными для использования кубитами (рис. 3.2).

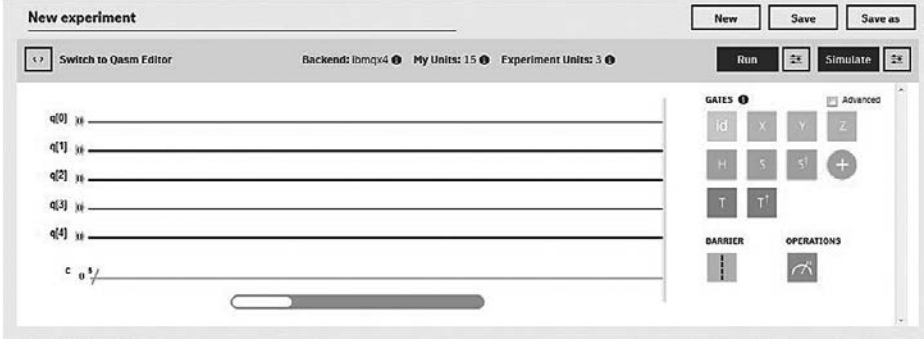

Рис. 3.2. Окно эксперимента в Composer

- В левой части гистограммы видны пять кубитов, доступных из процессора ibmqx4. Все они инициализированы в основное состояние $|0\rangle$. Линия внизу — это линия измерения, где будут собраны результаты схемы. Помните, что измерение должно выполняться последним в схеме, так как все операции вентилей выполняются параллельно и с наложением состояний.
- В правой части — квантовые вентили. Перетащите их к местоположению определенного кубита на гистограмме, чтобы начать строить схему.

Квантовые вентили

Квантовые вентили, поддерживаемые IBM Q Experience, описаны в табл. 3.1.

Таблица 3.1. Квантовые вентили для IBM Q Experience

Вентиль	Описание
Паули X 	Вращает кубит на 180° вокруг оси X. Преобразует $\|0\rangle$ в $\|1\rangle$ и $\|1\rangle$ — в $\|0\rangle$. Известен также как инвертирование разрядов или НЕ-вентиль. Представлен матрицей $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Вентиль	**Описание**
Паули Y 	Производит поворот вокруг оси Y сферы Блоха на π радиан. Представлен матрицей Паули $Y = \begin{bmatrix} 0 & -i \\ -i & 0 \end{bmatrix}$, где $i = \sqrt{-1}$ — мнимая единица
Паули Z 	Производит поворот вокруг оси Z сферы Блоха на π радиан. Представлен матрицей Паули $Z = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}$
Адамара 	Представляет собой поворот на π радиан вокруг оси $(X+Z)/\sqrt{2}$. Иначе говоря, преобразует состояния: • $\|0\rangle$ в $(\|0\rangle+\|1\rangle)/\sqrt{2}$; • $\|1\rangle$ в $(\|0\rangle-\|1\rangle)/\sqrt{2}$. Этот вентиль требуется для создания суперпозиций
Фазового сдвига $\sqrt{Z}$ 	Обладает свойством преобразования $X \to Y$ и $Z \to Z$. Данный вентиль расширяет вентиль Адамара для создания сложных суперпозиций
Эрмитово-сопряженная матрица для вентиля фазового сдвига $\sqrt{Z}$ 	Осуществляет преобразования $X \to -Y$ и $Z \to Z$
Управляемое НЕ (CNOT) 	Двухкубитный вентиль, который инвертирует целевой кубит (применяет оператор Паули X), если управляющий находится в состоянии 1. Этот вентиль требуется для создания запутывания
Фазового сдвига $\sqrt{S}$ 	Вентиль $\sqrt{S}$ выполняет половину обмена состояниями между двумя кубитами. Является универсальным: любой квантовый мультикубитный вентиль можно построить только из вентилей типа sqrt(swap) и однокубитных вентилей. Он представлен матрицей: $\sqrt{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2(1+i) & 1/2(1-i) & 0 \\ 0 & 1/2(1-i) & 1/2(1+i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Продолжение ⇩

Таблица 3.1 *(продолжение)*

Вентиль	Описание
Эрмитово-сопряженная матрица для вентиля фазового сдвига $\sqrt{S}$ или оператор ИЛИ-НЕ, примененный к вентилю фазового сдвига $\sqrt{S}$ 	Представлен матрицей $\sqrt{S} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1/2(1-i) & 1/2(1+i) & 0 \\ 0 & 1/2(1+i) & 1/2(1-i) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
Барьера 	Предотвращает преобразования вдоль его исходной линии
Измерительный 	Принимает на вход кубит в суперпозиции состояний и выдает либо 0, либо 1. Кроме того, вывод не является случайным. Есть вероятность того, что вывод примет значение 0 или 1, которое зависит от первоначального состояния кубита
Условный 	Применяет квантовую операцию при выполнении условия
Физическое частичное вращение (вентили U) 	U1 — однопараметрический однокубитный вентиль. U2 — однокубитный двухпараметрический одноимпульсный вентиль. U3 — однокубитный трехпараметрический двухимпульсный вентиль
Единичный (identity) 	Выполняет операцию простоя (idle) на кубите в течение одной единицы времени

Вы можете перетаскивать вентили из правой части Composer, чтобы создать схему, а если предпочитаете писать код на ассемблере, переключитесь в режим редактора QASM (рис. 3.3).

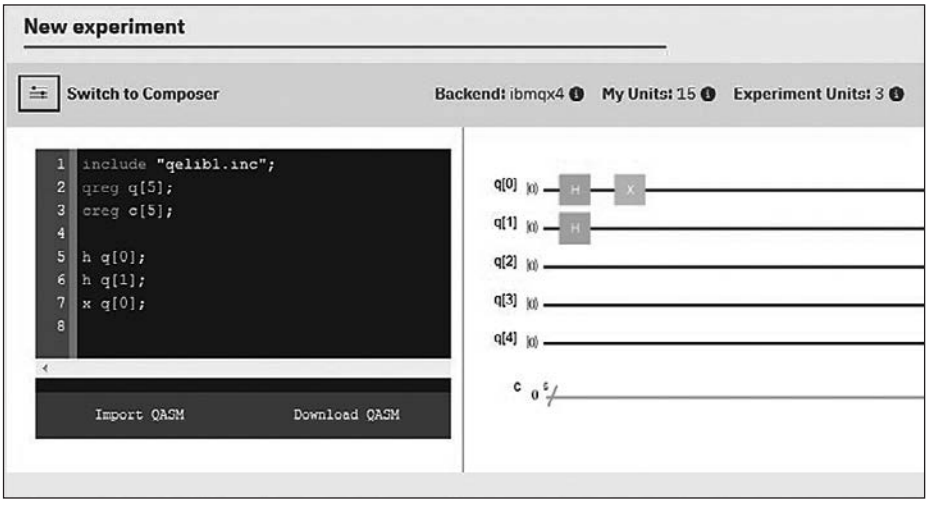

Рис. 3.3. Редактор экспериментов в режиме QASM

ПРИМЕЧАНИЕ

QASM — это квантовый язык ассемблера, созданный на базе платформы OPENQASM. Используется для экспериментов с неглубокими квантовыми схемами. Несмотря на то что умение писать на ассемблере стало чем-то вроде утраченного искусства, некоторым QASM может показаться более удобным, чем Python SDK или даже визуальный редактор.

Теперь рассмотрим различные квантовые процессоры.

Доступное квантовое серверное ПО

Есть несколько квантовых процессоров, которые можно выбрать для проведения экспериментов. В табл. 3.2 приведен официальный список, ранжированный в соответствии с количеством кубитов (по информации с сайта о серверном ПО для IBM Q Experience).

Таблица 3.2. Официальный список квантового серверного ПО, доступного для пользователей IBM Q Experience

Наименование	Подробности
Ibmqx2	Условное название: Sparrow. Количество кубитов: 5. Доступен онлайн с 24 января 2017 года
Ibmqx4	Условное название: Raven. Количество кубитов: 5. Доступен онлайн с 25 сентября 2017 года
Ibmqx3	Условное название: Albatross. Количество кубитов: 16. Доступен онлайн с июня 2017 года
Ibmqx5	Условное название: Albatross. Количество кубитов: 16. Доступен онлайн с 28 сентября 2017 года

В табл. 3.2 приведен официальный список процессоров, доступных на момент написания книги, но существует куда более интересный способ получить обновленный список доступных машин в реальном времени с помощью превосходного REST API. Более подробное описание этого API дано в разделе «Удаленный доступ через REST API» текущей главы, а пока продемонстрирую, как получить всегда актуальный список серверного ПО, используя конечную точку REST *Available Backend List*: https://quantumexperience.ng.bluemix.net/api/Backends?access_token=ACCESS-TOKEN.

СОВЕТ

Чтобы получить токен доступа, смотрите подраздел «Аутентификация» раздела «Удаленный доступ через REST API» на с. 130. Обратите внимание, что токен API не совпадает с токеном доступа. Токены API используются для выполнения квантовых программ через Python SDK. Токены доступа применяются для вызова REST API.

Перейдя по URL, приведенному в предыдущем абзаце, вы увидите список квантовых процессоров в формате JSON. Вот как это выглядит на момент написания книги (листинг 3.1). Учтите, что полученный вами результат может отличаться.

Листинг 3.1. HTTP-ответ на вызов Backend Information REST API

```
[{
    "name": "ibmqx2",
    "version": "1",
    "status": "on",
    "serialNumber": "Real5Qv2",
    "description": "5 transmon bowtie",
    "basisGates": "u1,u2,u3,cx,id",
    "onlineDate": "2017-01-10T12:00:00.000Z",
    "chipName": "Sparrow",
    "id": "28147a578bdc88ec8087af46ede526e1",
    "topologyId": "250e969c6b9e68aa2a045ffbceb3ac33",
    "url": "https://ibm.biz/qiskit-ibmqx2",
    "simulator": false,
    "nQubits": 5,
    "couplingMap": [
        [0, 1],
        [0, 2],
        [1, 2],
        [3, 2],
        [3, 4],
        [4, 2]
    ]
}, {
    "name": "ibmqx5",
    "version": "1",
    "status": "on",
    "serialNumber": "ibmqx5",
    "description": "16 transmon 2x8 ladder",
    "basisGates": "u1,u2,u3,cx,id",
    "onlineDate": "2017-09-21T11:00:00.000Z",
    "chipName": "Albatross",
    "id": "f451527ae7b9c9998e7addf1067c0df4",
    "topologyId": "ad8b182a0653f51dfbd5d66c33fd08c7",
    "url": "https://ibm.biz/qiskit-ibmqx5",
    "simulator": false,
    "nQubits": 16,
    "couplingMap": [
        [1, 0],
        …
```

```
        [15, 14]
        ]
}, {
    "name": "Device Real5Qv1",
    "status": "off",
    "serialNumber": "Real5Qv1",
    "description": "Device Real5Qv1",
    "id": "cc7f910ff2e6860e0d4918e9ee0ebae0",
    "topologyId": "250e969c6b9e68aa2a045ffbceb3ac33",
    "simulator": false,
    "nQubits": 5,
    "couplingMap": [
        [0, 1],
        [0, 2],
        [1, 2],
        [3, 2],
        [3, 4],
        [4, 2]
    ]
}, {
    "name": "ibmqx_hpc_qasm_simulator",
    "status": "on",
    "serialNumber": "hpc-simulator",
    "basisGates": "u1,u2,u3,cx,id",
    "onlineDate": "2017-12-09T12:00:00.000Z",
    "id": "084e8de73c4d16330550c34cf97de3f2",
    "topologyId": "7ca1eda6c4bff274c38d1fe66c449dff",
    "simulator": true,
    "nQubits": 32,
    "couplingMap": "all-to-all"
}, {
    "name": "ibmqx4",
    "version": "1",
    "status": "on",
    "serialNumber": "ibmqx4",
    "description": "5 qubits transmon bowtie chip 3",
    "basisGates": "u1,u2,u3,cx,id",
    "onlineDate": "2017-09-18T11:00:00.000Z",
    "chipName": "Raven",
    "id": "c16c5ddebbf8922a7e2a0f5a89cac478",
    "topologyId": "3b8e671a5a3b56899e6e601e6a3816a1",
    "url": "https://ibm.biz/qiskit-ibmqx4",
    "simulator": false,
    "nQubits": 5,
    "couplingMap": [
        [1, 0],
        [2, 0],
        [2, 1],
```

```
        [2, 4],
        [3, 2],
        [3, 4]
    ]
}, {
    "name": "ibmqx3",
    "version": "1",
    "status": "off",
    "serialNumber": "ibmqx3",
    "description": "16 transmon 2x8 ladder",
    "basisGates": "u1,u2,u3,cx,id",
    "onlineDate": "2017-06-06T11:00:00.000Z",
    "chipName": "Albatross",
    "id": "2bcc3cdb587d1bef305ac14447b9b0a6",
    "topologyId": "db99eef232f426b45d2d147359580bc6",
    "url": "https://ibm.biz/qiskit-ibmqx3",
    "simulator": false,
    "nQubits": 16,
    "couplingMap": [
    …
    ]
}, {
    "name": "QS1_1",
    "version": "1",
    "status": "standby",
    "serialNumber": "QS1_1",
    "description": "20 qubit device v1",
    "basisGates": "SU2+CNOT",
    "onlineDate": "2017-10-20T11:00:00.000Z",
    "chipName": "Qubert",
    "id": "cb141f7bb641b8a10487a6fab8483b86",
    "topologyId": "25197b9b73c4b52ca713ca4d126417b5",
    "simulator": false,
    "nQubits": 20,
    "couplingMap": [
    …
    ]
}, {
    "name": "ibmqx_qasm_simulator",
    "status": "on",
    "description": "online qasm simulator",
    "basisGates": "u1,u2,u3,cx,id",
    "id": "18da019106bf6b5a55e0ef932763a670",
    "topologyId": "250e969c6b9e68aa2a045ffbceb3ac33",
    "simulator": true,
    "nQubits": 24,
    "couplingMap": "all-to-all"
}]
```

В листинге 3.1 показан текущий список доступных процессоров, который по большей части совпадает с официальным перечнем с сайта IBM Q Experience. Тем не менее там есть много дополнительной интересной информации о конструктивных схемах машин.

- Дополнительные процессоры и симуляторы.
  - Похоже, что для использования доступны два удаленных симулятора — `ibmqx_qasm_simulator` и `ibmqx_hpc_qasm_simulator`, хотя в официальной документации упоминается только `ibmqx_qasm_simulator`. Эта информация может пригодиться при тестировании сложных схем: чем больше симуляторов, тем лучше.
  - Уже давно ходят слухи о 20-кубитном процессоре. Поговаривают даже о запланированном выходе 50-кубитного монстра к концу 2018 года[1]. Этот список, по-видимому, подтверждает по крайней мере существование 20-кубитной машины. Но пока рано радоваться, она будет доступна только для корпоративных клиентов.
- Помимо обычной информации, такой как название машины, версия, состояние, количество кубитов и т. д., есть термины, с которыми мы должны ознакомиться.
  - *basisGates* — физические кубитные вентили процессора. Это основа, на которой могут быть построены более сложные логические элементы. Большинство процессоров в списке используют u1, u2, u3, cx, id:
    - вентили u1, u2, u3 называются *частичными НЕ-вентилями*, они вращают кубит вокруг осей *X*, *Y*, *Z* на θ, ϕ или λ радиан;
    - cx называется *управляемым НЕ-вентилем* (CNOT или CX). Он задействует два кубита и выполняет операцию НЕ на втором кубите, только когда первый кубит находится в состоянии $|1\rangle$, и оставляет его неизменным в противном случае;
    - id — это единичный вентиль, который выполняет операцию простоя (idle) на кубите в течение одной единицы времени.

[1] IBM уже предоставляет доступ к Q System One с 20 кубитами и в ближайшее время откроет для партнеров доступ к квантовому компьютеру с 50 кубитами. — *Примеч. ред.*

- *couplingMap* — карта связей. Определяет взаимодействия между отдельными кубитами, сохраняя при этом квантовую когерентность (или чистое состояние — представьте, что солдаты, переходя реку по старому мосту, идут не в ногу, чтобы амплитуды их шагов не совпали и это не привело к разрушению моста). Связывание кубитов в пары используется для упрощения квантовой схемы и позволяет разбить систему на более мелкие единицы.

Теперь вернемся к Composer, чтобы создать первую квантовую композицию.

Опус 1: вариации на тему состояний Белла и GHZ

Здесь мы рассмотрим два умопомрачительных квантовых эксперимента, используемых для демонстрации странности квантовой механики:

- *состояния Белла* — показывают, что физика не описывается локальной реальностью. Это то, что Эйнштейн назвал *мистическим дальнодействием*;
- *GHZ-состояния* — даже еще более странные, чем состояния Белла, GHZ-состояния (названные в честь своих создателей Гринбергера, Хорна и Цейлингера) являются трехкубитным обобщением состояний Белла.

Рассмотрим их подробнее.

Состояния Белла и мистическое дальнодействие

Состояния Белла являются экспериментальной проверкой известных неравенств Белла. В 1964 году ирландский физик Джон Белл предложил способ проверки квантовой запутанности (мистического дальнодействия). Он вывел ряд неравенств, которые стали очень востребованы в физическом сообществе. Они известны как неравенства Белла (сегодня называют теоремой Белла).