

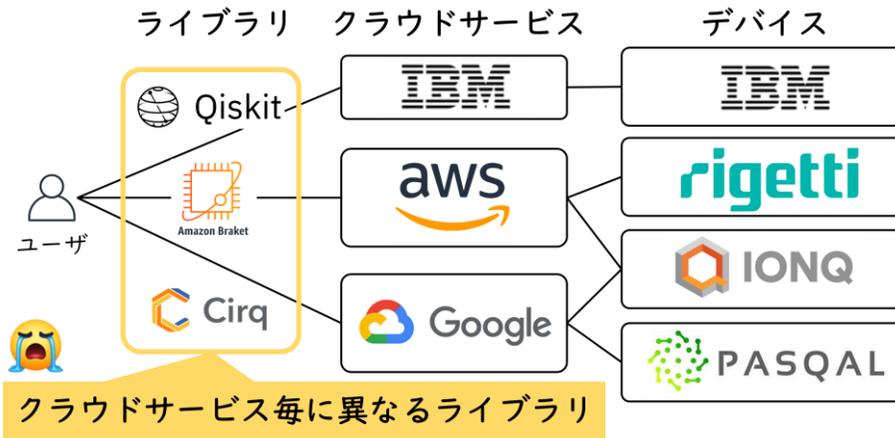
オープンソースの量子コンピュータ・クラウド基盤開発

— 多様なユーザと多様な量子コンピュータをつなぐ —

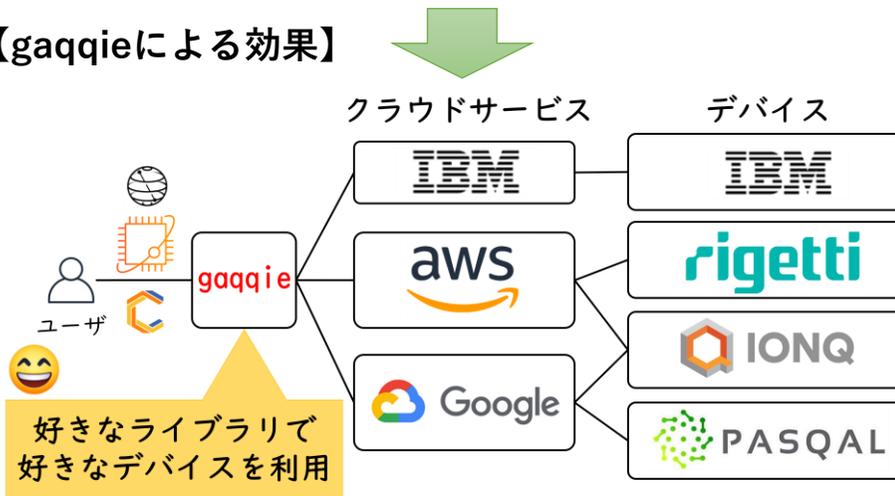
東野仁政

Gate-based Quick Quantum Infrastructure (gaqqie, ガッキー) <https://github.com/gaqqie/gaqqie>

【量子コンピュータ・クラウドサービスの現状】



【gaqqieによる効果】



【多様なユーザに対するメリット】

一般のユーザ

好きなライブラリで実装、好きなデバイスで実行
量子計算の研究室

さまざまなデバイスでの実行結果を一元管理
独自のデバイスも追加可能

量子コンピュータ・デバイスの研究開発者

少ない手間でクラウドサービスを公開
セキュリティにも強く低コストで運用可能

【gaqqieのメリット】

オープンソース

量子コンピュータのクラウドサービスでは世界初
仕様やノウハウが公開されている

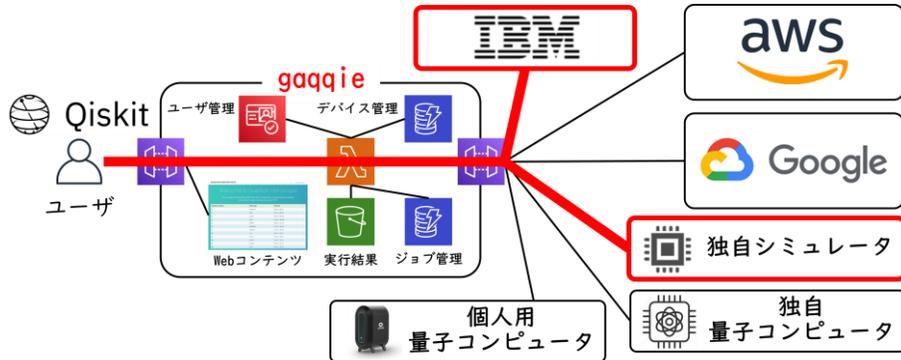
サーバレス・アーキテクチャ

耐障害性、スケーラビリティ、低コストでの運用

クラウド～デバイス間でプル型の通信が可能

多数のジョブを受信してもデバイスは低負荷

【IBMと独自シミュレータで実行可能(2021年2月時点)】



異なるサービスの実行結果をまとめて表示する画面

The screenshot shows a dashboard with a table of job execution results. Callouts indicate that pending jobs can be cancelled and that results from IBM and independent simulators are summarized. A 'Download' button is visible for each row.

Can	Job ID	Status	Provider	Device	Create Time	End Time	Download
🚫	33ff1afc-a3b3-4959-85f9-22a312b1642d	QUEUED	gaqqie	qiskit_simulator	2022-02-08 20:47:20		
	b42b0abb-88d9-4842-899e-7ec268674d54	SUCCEEDED	gaqqie	qiskit_simulator	2022-02-08 14:06:03	2022-02-08 14:06:16	📄
	5d7d8016-aa54-4cd0-9d16-94ded1563fc9	SUCCEEDED	gaqqie	qiskit_simulator	2022-02-08 13:41:25	2022-02-08 13:41:37	📄
	ffe3f9f7-1d1d-4cb3-8e9a-00b5717955c7	SUCCEEDED	IBM	ibmq_quito	2022-02-08 07:22:12	2022-02-08 07:30:48	📄
	a042e97e-dbe4-48a4-817e-23b7f2963c64	SUCCEEDED	IBM	ibmq_quito	2022-02-08 04:09:29	2022-02-08 07:07:06	📄

gaqqieを簡単に利用できる仕組み

【ユーザ側の実装サンプル】

```
from qiskit import QuantumCircuit, execute
from gaqqie_door import QiskitGaqqie
```

ライブラリをインポート

```
circuit = QuantumCircuit(2, 2)
circuit.h(0)
circuit.cx(0, 1)
circuit.measure([0, 1], [0, 1])
```

Qiskitの経験者にとって自然な書き方

```
url = "https://<api-id>.execute-api.<region>.amazonaws.com/<stage>"
QiskitGaqqie.enable_account(url)
backend = QiskitGaqqie.get_backend("qiskit_simulator")
```

デバイスを指定

```
job = execute(circuit, backend)
result = job.result()
print(f"result job_id={job.job_id()}, counts={result.get_counts()}")
```

【デバイス側の実装サンプル】

```
url = "https://<api-id>.execute-api.<region>.amazonaws.com/<stage>"
app = Gaqqie(url)

@app.receive_job(device_name="qiskit_simulator", interval=10)
def receive_job(job):
    # parse circuit
    ...

    # execute circuit
    ...
    result = aer_job.result()
    print(f"result job_id={job_id}, counts={result.get_counts()}")
    result_dict = result.to_dict()
    result_dict["backend_name"] = job.device_name
    result_json = json.dumps(result_dict, indent=2)

    # register result
    job_result = Result(job_id=job_id, status="SUCCEEDED", results=result_json)
    response = app.register_result(job_result)
```

デコレータ(@app.~)を付与しておけばクラウドサービスがジョブを受信したときにデバイス側で動作する関数が呼び出される

実行結果をクラウドに登録

```
app.join()
```