note 14 views

Command line script for submitting jobs to the server

Hi all, I was working on submitting jobs to the server and to make it easier I have created a script that can do it from the command line.

The script works by first checking the code to see if it is valid, and then it submits it to the server if the checks parse. The use case is like this:

```
./post.py myemail@address.com --shots 10 program1.quil program2.quil
```

If you are happy about the checks and the measurements you can commit it to the server

```
./post.py myemail@address.com --commit --shots 10 program1.quil program2.quil
```

You can also skip the checks with `--no-check` if you do not have quil installed on your system.

If everything go as planned you should see a "`{'message': 'saved to DB'}`" for each program, and receive an email within minutes saying something on the line of

```
QVM SIMULATION:
 Your job has been verified to run during the next reservation period
```

Hope this helps!

Here is the script (and for easy download post.py)

```
#!/usr/bin/env python3
from pathlib import Path
import pprint
import time

pp = pprint.PrettyPrinter(indent=4)

class Timer:
    def __enter__(self):
        self.start = time.time()

    def __exit__(self, exc_type, exc_val, exc_tb):
        self.end = time.time()

    def __str__(self):
        return (f"{self.end - self.start:0.2f} s")


def main(args):
    jobs = { str(f) : f.read_text() for f in args.program }
    print(f"Found {len(jobs)} job(s).")

    if not args.no_check:
        for key, program in jobs.items():
            print(f"Checking {key}...")
            t = Timer()
            with t:
                measure = check_program(program, args.shots, args.lattice)
            pp.pprint(measure)
            print(f"Check successfull! [{t}]")

    if args.commit:
        send_jobs(jobs, args.email, args.server, args.shots)

def check_program(program, shots, lattice):
    from pyquil import get_qc
    from pyquil.parser import parse_program
    from pyquil.api import local_qvm

    p = parse_program(program)
    with local_qvm():
        qc = get_qc(lattice, as_qvm=True)
        return qc.run_and_measure(p, trials=shots)

def send_jobs(jobs, email, server, shots):
    """ Sends jobs to the server """
    import requests
    import json

    for key, program in jobs.items():
        print(f"Sending {key}...")
        r = requests.post(server + "/send", data={
            'quil': program,
            'email': email,
            'shots': shots
            })
        print(f"Result {r}")
        obj = json.loads(r.text)
        pp.pprint(obj)


if __name__ == "__main__":
    import argparse
    parse = argparse.ArgumentParser(
            description="Submit a job to the server",
            formatter_class=argparse.ArgumentDefaultsHelpFormatter
            )
    parse.add_argument('email', metavar="EMAIL",
            help='the email to send the results to')
    parse.add_argument('program', metavar="PROGRAM", type=Path, nargs='+',
            help='quil program files')
    parse.add_argument('--shots', default=1, type=int,
            help='number of shots')
    parse.add_argument('--server', default="http://63.110.29.106:80",
            help='the server to post job to')
    parse.add_argument('--lattice', default="Aspen-4-6Q-A",
            help='the lattice to check against')
    parse.add_argument('--commit', action='store_true',
            help='send the request to the server')
    parse.add_argument('--no-check', action='store_true',
            help='do not check the program before sending to server')

    main(parse.parse_args())
```

logistics

Updated 7 hours ago by Christian Gram Kalhauge

followup discussions *for lingering questions and comments*