Quantum Software Engineering notes

Leonardo Gori, Stefano Taborelli

March 30, 2022

1 Introduction

These notes are made in order to maintain a sort of logbook of our introduction to the quantum computing's literature. The structure and the content of this document is, in the moment, incomplete, and therefore it is supposed to be updated during time. In the meanwhile the next version of the document is committed, the temporary structure is composed by two sections. Section 2 treats the argument of quantum computing from a informative point of view, while in section 3 technical information about quantum information theory are supposed to be added.

2 Scientific dissemination's perspective

From the scientific dissemination's perspective, we are aware about the computational power potentialities of quantum computing devices. As seen in the course we are attending, the capability of these devices allowed the introduction of the class of problems called BQP (Bounded-error Quantum Polynomial time) that groups those problems that are solvable by a quantum computer in polynomial time with a probability of error limited by 1/3.

However, when dealing with such technologies for the development of software services, there are some drawbacks that are necessary to be considered in real world. According to [Mog+20], we provide here some of the main complexities that should be faced, in order to spread the use of quantum technologies in ordinary life:

- Deployment complications: since the physical components that build up quantum systems need to be put in special environments, in order to avoid problems like *decoherence* and interference due to noise, the deployment on ordinary personal devices is actually one of the main challenges in quantum software engineering
- the costs needed for the hardware are very high
- Platform dependencies: as well as for the beginning of the classical computation era, where programmers were used to develop solutions tightly

linked to the specifics of the hardware. Nowadays, with the limitations in qubits of current hardware, they develop mostly experiments for a specific machine and show how small, very concrete problems can be solved by a quantum computer

- niche sector: programming languages available for QC are still essentially low-level, operating at the level of quantum gates ([Gom+20]). This can represent a problem, since the development of algorithms is tightly dependent on the developer knowledge about quantum computation.
- Verification and validation of software solutions: as mentioned in [Gom+20], due to the fact that quantum states collapse when measured, leads to the problem of accessing the state of the computation of a program for verification purposes, without inherently destroying superposition

3 Technical perspective

...

References

[Gom+20] Cláudio Gomesa et al. “Off-the-shelf Components for Quantum Programming and Testing”. In: Oct. 2020.

[Mog+20] Enrique Moguel et al. “A Roadmap for Quantum Software Engineering: applying the lessons learned from the classics”. In: Oct. 2020.