Easy access to quantum computing for applications

PD Dr. Jeanette Lorenz, Fraunhofer-Institute for Cognitive Systems IKS

Quantum computing can result in disruptive changes in many industrial areas, e.g. through the more efficient solution of optimization problems e.g. in logistics. To profit from a potential quantum advantage fully, an easy access to quantum computing hardware and software is required which does not require profound knowledge of the user regarding the underlying physics or technology. Within the Munich Quantum Valley, we develop a high-level software eco-system to achieve exactly this in a unique approach including all elements of the software stack. This includes e.g. the automatic decomposition of application problems into parts better suited for classical, HPC-systems or quantum computers, the development of mapping libraries to map the resulting software optimally onto the quantum computing hardware, down to the co-design of software and hardware. At the same time different tools ensure robust algorithms to solve the application problem.

Quantum computing is predicted to result in disruptive changes in many industrial sectors, e.g. by more efficiently solving optimization problems in logistics or in production, or in the areas of artificial intelligence and simulation problems, which e.g. are relevant in the pharma sector in the design of new drugs. However, to realize these potential benefits, significant progress in quantum computing is still required. This does not only concern improvements in the quantum hardware with the need of more qubits of higher quality and noise correction. Also, the full software stack needs to be developed in an easily accessible way for the end user. Quantum computers need to be integrated with classical systems including HPC-systems to ensure that a computational task is using the best suited systems for a problem at hand.

The Munich Quantum Valley is a new initiative in Garching/Munich (Germany) with the ambitious aim to build multiple quantum computers based on different technologies along with a full software stack up to (industrial) applications. This includes – besides the construction of the quantum computers – also their integration into existing computing infrastructure including HPC-systems, as well as the software that is required to realize the solution of application problems on quantum computers.

Despite huge progress in realizing programming interfaces for quantum computers, the design of quantum algorithms is still very uncommon for the typical industrial software engineer – even if the API is embedded in high-level languages like Python. There are a couple of initiatives to build quantum-computing specific high-level languages – IBM Qiskit is certainly most prominent and widely used. These high-level languages are however not yet optimized in all aspects, and most importantly still require profound knowledge about quantum computing of the end user. In general, quantum software development still needs to happen relatively close to the quantum hardware. It needs to answer such questions like: which quantum hardware is best suited for a specific application problem? How can one map a specific quantum algorithm best on the native architecture of the then chosen quantum-hardware? How can one guarantee that the quantum algorithm and quantum computer really delivers robust and reliable results?

In our local projects we aim to exactly solve these shortcomings. Our vision is that the (industrial) end user should only be required to have minimal knowledge about quantum computing hardware and quantum computing software, but the user is still able to profit in an easily accessible and reliable way from quantum-assisted solutions of the application problem. And this by using the quantum hardware in as efficient way as possible. Eventually this will imply that also the unexperienced user can profit as early as possible from the potential benefits that quantum computers bring for real-life applications.

To realize this vision, we take artificial intelligence as model. There are a couple of core points why artificial intelligence and machine learning is widely used with significant added value nowadays. These are the availability of large amounts of training data, the availability of the required computational resources including cloud services, and most importantly the existence of an eco system consisting of easily usable software tools and libraries such as e.g. Scikit-Learn, Tensorflow or PyTorch. Taking all this together, the entry barrier to use and apply artificial intelligence was significantly reduced and a transfer of the technology from academia to (industrial) applications was feasible.

Along these lines, we would like to build a quantum-computing-related eco system. Related project proposals were recently funded (https://www.munich-quantum-valley.de/) or are in the final stage of the review process of German federal funding. In our developments we start from a wide range of industrial use cases, e.g. within the area of optimization problems. After having identified the commonalities of these problems, we will develop high-level libraries which will automatically split the original problem into sub-problems in a quantum-hardware-agnostic way. This will create an additional abstraction layer which is necessary to allow users an easy access in their familiar programming environment. These libraries will include decomposition tools and methods that automatically split the problem in parts computationally better suited for quantum computers, classical systems or HPC-systems, along with a performance analysis of the solution. Additionally, methods are developed to split the resulting quantum algorithms into smaller parts and algorithms so that one can use the currently available Noisy-Intermediate-Scale Quantum (NISQ) hardware in the most efficient way. (De-)encoding strategies then help to map the (classical) information in the best possible way into quantum computers. These smaller parts are again then mapped in an optimized way via a library consisting of mappings and algorithmic building blocks onto the hardware, including co-design developments to benefit from quantum hardware fully. Quantum computing will only be useful for applications if quantum algorithms can also be used reliably. To this extend, we will develop an evaluation and verification procedure.

The currently available NISQ-hardware requires the use of hybrid algorithms, where a close interaction between classical computers, HPC-systems and quantum computers is needed. It is essential that this interaction happens in a fast way with short latency, so that quantum-assisted solutions are beneficial for real-life applications. This can be realized through a direct or virtualized environment on the control system of quantum computer with a direct communication channel to the quantum computer. This requires novel concepts for run time environment with additions to the typical mono-lithic total-job approach. Additionally, a flexible data environment is needed to communicate parameters and measurements between the systems, e.g. via data-staging and a data management in a hybrid system similar to OpenMP and OpenCL in the HPC-context.

Taking all these developments together, we will create a unique software eco-system within the Munich Quantum Valley which will ease the use of quantum computing for the end user. In this talk, I will report and highlight our plans and our visions.