IQM

Leibniz-Rechenzentrum
der Bayerischen Akademie der Wissenschaften

2022

Bringing quantum acceleration to supercomputers

MARTIN RUEFENACHT[1], BRUNO G. TAKETANI[2], PASI LÄHTEENMÄKI[3], VILLE BERGHOLM[3], DIETER KRANZLMÜLLER[1], LAURA SCHULZ[1], AND MARTIN SCHULZ[1,4]

[1] *Leibniz Supercomputing Centre, Garching near München, Germany*
[2] *IQM, Nymphenburgerstr. 86, 80636 Munich, Germany*
[3] *IQM, Keilaranta 19, 02150 Espoo, Finland*
[4] *Technical University of Munich, Garching near München, Germany*

Abstract

In a world shifting towards sustainable growth, high-performance computing has an important challenge: delivering on a growing demand for increased computational power, while keeping energy consumption at bay. Quantum computers promise to meet these challenges with an exponential performance improvement for key applications and are anticipated to be the next big technological breakthrough in the field. This paper discusses part of the road ahead to integrate quantum acceleration into supercomputers, as well as the critical steps and decisions required in order to build the quantum future of high-performance computing and make important strides towards green computing.

Introduction

The digital revolution has greatly changed our way of life by significantly increasing our computational capabilities and by democratizing access to it. Be it finding the route to our next appointment, driving to it in a high-speed train or simply checking the weather before departure, we rely on complex computer calculations mostly accessed over the internet. In this landscape, supercomputing centres and data centres play a central role by hosting considerable computing resources that can be accessed remotely by their users. On the other side, enterprises often host their private high-performance computing (HPC) systems for sensitive workflows and cost optimization.

Unsurprisingly, this increased demand has fuelled the rapid development and deployment of new computing resources such as peta-, pre-exa-, and exa-scale supercomputers, as well as new hardware accelerators such as general-purpose graphics processing units (GPUs), field-programmable gate arrays (FPGAs), application-specific integrated circuit (ASICs) and others (see Figure 1. This specialized hardware is designed to execute specific functions more efficiently than other systems, such as central processing units (CPU)), whose architecture is tailored for general-purpose usage.

The efficiency of hardware acceleration also helps tackle a critical challenge of the industry: energy consumption. Supercomputers use a significant amount of power, with the Top500 HPC systems estimated to need over 650MW combined when operating at peak capacity [12][1]. Accelerators are key in designing more energy efficient systems to enable exa-scale supercomputers.

Springing out of the physics laboratories, quantum computers (QC) are one of the most promising accelerators as they present an inherently different computing paradigm. Other accelerators, most notably GPUs, draw their improved performance from their specialized design and/or increased parallelism, but ultimately rely on the same computing paradigm (typically "von Neumann"-based), or at least technology (typically CMOS-based) as general-purpose hardware.

Contrary to these, quantum computers allow for a fundamental change in how problems are formulated, reducing the complexity class of several key applications [14]. This change is at the root of their potential advantage.

Quantum computers encode information in the state of quantum bits (qubits) and use external signals (e.g., via microwaves or lasers) to manipulate them. Using the properties of quantum physics, these can be used by quantum algorithms to achieve exponential improvements on resource scaling[2]. Several such quantum algorithms have already been developed [11]; nevertheless, it is important to highlight that QCs are not foreseen as a replacement of existing computing technology. Quantum computers are more suitable to tackle problems for which the classical computing resources needed scale exponentially with the problem size. Other problems are likely to see smaller or even no benefit from quantum computers and the same is true for auxiliary tasks, like pre- and post-processing, I/O and visualization. This understanding helps situate QCs in the plethora of computing hardware as an accelerator to existing high-performance computing systems, specifically adapted to certain classes of problems, for which quantum computers will be a disruptive technology.

[1] Estimate taken by extending the available power efficiency data to be representative of the full list.
[2] For a detailed introduction to quantum computing the reader is encouraged to read reference [14]

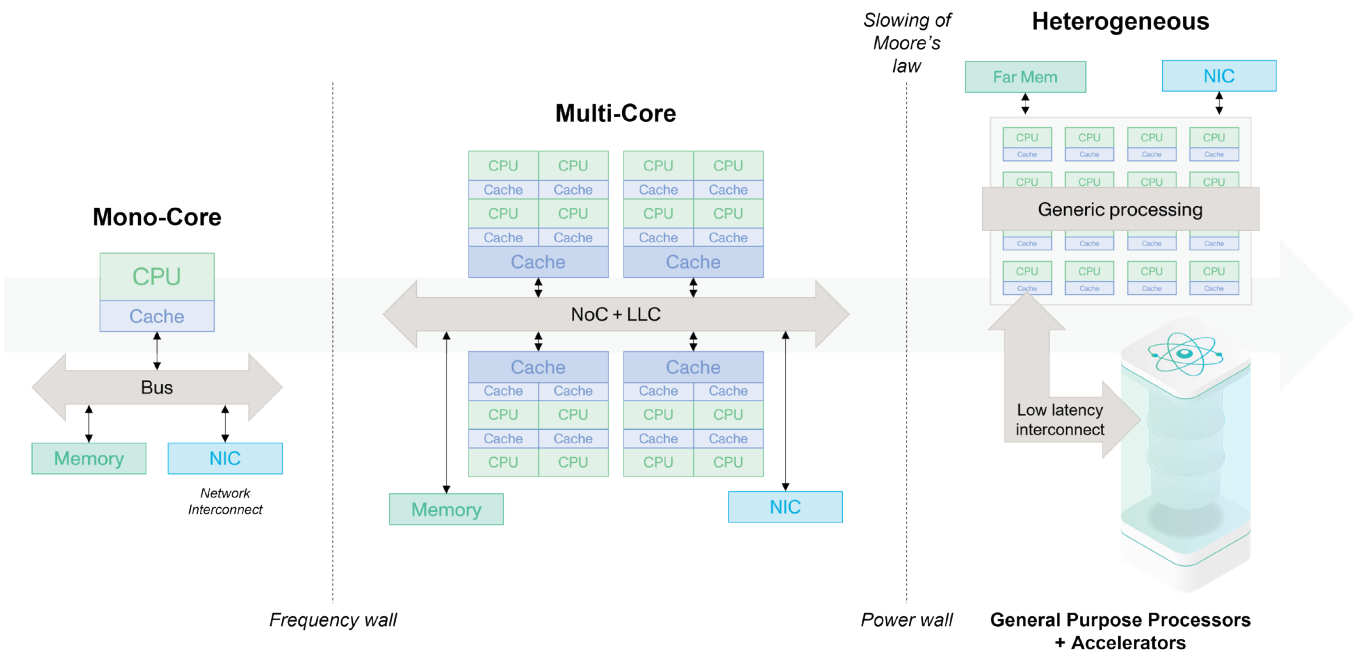


FIGURE 1. COMPUTE ARCHITECTURE EVOLUTION

Important development walls faced in the evolution of computing systems and the architectural solutions overcoming these challenges.

Image source: European Processor Initiative (EPI), and Sipearl.

From the above perspective, it is already natural to envision quantum computers as an important accelerator in HPC systems. This concept is further enhanced by acknowledging that many applications of quantum computing align with research domains with heavy HPC usage. The simulation of quantum systems, such as for chemistry or materials research, optimization problems, and solutions to differential equations, e.g., for hydrodynamics or financial predictions, are some of the key areas where quantum algorithms have been identified [11]. In practice, these quantum algorithms typically make up part of larger computations running on HPC systems. As such, the coordination of the complete hybrid computation between the different classical and quantum computing nodes is essential. Management of these resources becomes even more critical with the understanding that future hybrid systems will likely have multiple quantum accelerators, possibly with different specifications. These specifications can include, e.g., the number of qubits, their coherence times and gate fidelities, where coherence time is a measure related to how long the information stored in the qubits is reliable and gate fidelity relates to the quality of the operations (i.e., gates) that need to be performed. An ever-present concern with high-performance computing is the optimal usage of all resources, which involve minimizing time-to-solution and energy-to-solution of individual algorithms as well as of the global use of these resources. While quantum computing brings significant improvements to time and energy-to-solution, it presents new challenges to resource management which need to be addressed for the successful deployment of these technologies. Taking into account the specifications of different quantum computers may prove critical to optimise their individual and global usage.

The potential of quantum acceleration has been clearly identified by supercomputing centres, with a recent study showing that 76% of HPC centres worldwide plan to be using the technology by 2023, and that 71% plan to move to on-premises quantum computing by 2026 [9]. Moreover, several activities to foster the integration of these systems are ongoing or have been announced, noticeably by European countries [7, 8] and the European Commission [15, 16].

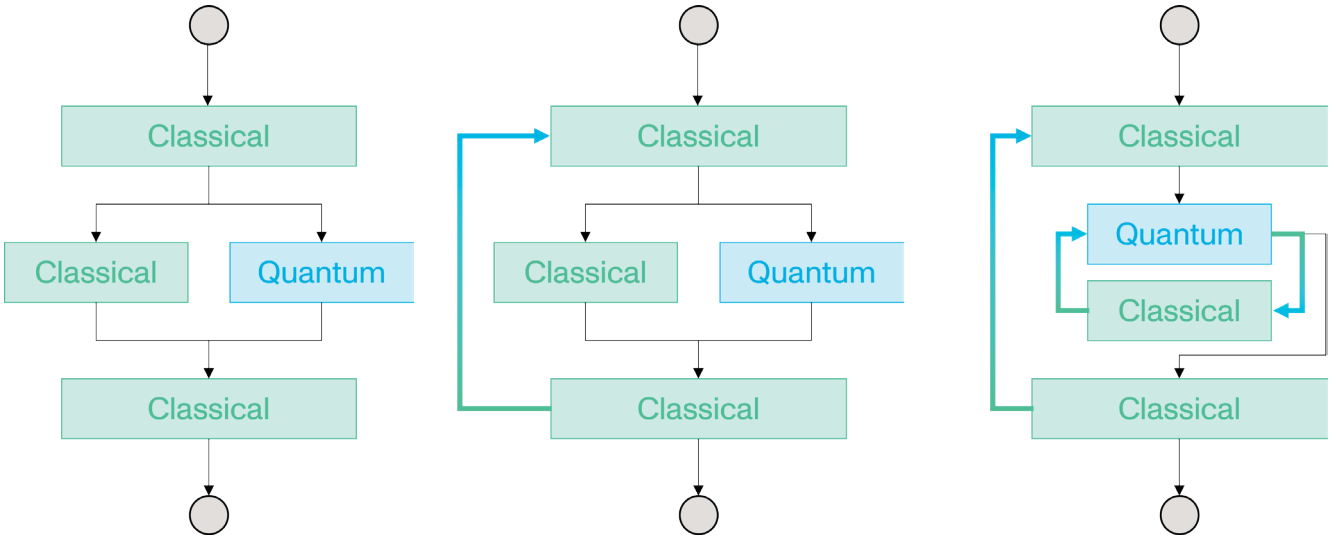
In this article, we will discuss the implications of integrating quantum computing as an accelerator to HPC and the impacts on computing approaches this has. The impact of the different deployment architectures on the performance of the hybrid classical-quantum system will also be analysed. With these considerations, we finally propose the concept of a system-wide quantum resource manager to facilitate integration with multiple different quantum computers and top-level quantum frameworks.

Revisiting Computing

Quantum advantage is typically understood as the scenario where quantum computers outperform classical computers for relevant industrial or scientific applications. This is the ultimate goal of quantum computing and a number of quantum algorithms have been theoretically demonstrated to have quantum advantage. An important and exciting realization is that only a small number of applications were explored so far, so further developments of quantum algorithms will likely show an increasing number of compute domains where quantum computing can be disruptive, sometimes with the full exponential increase in performance, but oftentimes with sub-exponential benefits. Most developments to date have focused on algorithms running only on quantum computers. Going further, combining classical and quantum computers is likely to yield a large number of algorithms providing quantum advantage. A well-known case of such hybrid approach is Variational Quantum Algorithms [4]. These algorithms investigate problems well suited to quantum computers and adapt them to run in a loop where the quantum computer helps to evaluate a cost function which is then optimized in a classical computer.

Quantum computing experts, typically with background in physics or mathematics, have driven the development of quantum algorithms. These are experts on the tool, in search for its use. This may be due to quantum algorithms fundamentally requiring a rethink of the approach to scientific computing. However, algorithm development has seen an increased interest from domain experts from the various disciplines where quantum computing is expected to be disruptive.

Integrating quantum computing into HPC requires a similar approach to other accelerators we have seen in the past. The classical host is the computer which requires tasks to be computed using the accelerator. The host offloads these tasks to the accelerator, which executes the tasks and returns the desired output. Offloading is a well-known usage of accelerators that shifts individual kernels to the accelerators so they can exploit those resources for their specialized superior capabilities, resulting in a reduction of the overall time-to-solution. As will be discussed later, quantum accelerators might also profit from onloading where part of the classical processing needed to facilitate the operation of the QC might benefit from running in the HPC system. Successful integration of a quantum accelerator into the HPC environment relies on the HPC user community learning to rethink their problems considering these new resources; and this rethinking must be done along several different directions:



(a) usage of quantum in workflow (b) iterative workflow (c) onloading workflow

FIGURE 2. HYBRID ALGORITHMS USING QUANTUM ACCELERATION in (a) linear and (b) iterative workflows, as well as (c) a workflow including onloading work.

RETHINKING THE USE OF DIRECT SOLVERS

Due to the inherent probabilistic nature of measurements in quantum mechanics, the quantum algorithm must be repeated several times and the statistics of the measurement outcomes provides the desired output of the algorithm. Because of this, quantum algorithms most resemble classical direct solvers, in the sense that intermediate results are not useful towards a solution of the problem, only the final result is the solution. Due to the lack of usage of direct solving methods in high performance computing our approach must be rethought and we must consider which parts of the scientific problems being tackled can be evaluated faster or more accurately by a quantum computer. At this point, all resources in a computation can be used to their maximum potential.

RETHINKING WORKFLOWS

The variety of problems faced by HPC users require various distinct workflows. Using quantum accelerators will call for users to revisit their workflows and understand if parts of it have the potential for quantum advantage. Mapping of a subroutine to a known quantum algorithm would be the simplest way to see this. Closer analyses could also identify bottlenecks with, e.g., exponential scaling of resources, and appropriate quantum algorithms could then be developed. Some workflows require a one-time process which could be efficiently performed on a quantum computer, while infeasible on a classical computer, as seen in Figure 2a. This implies a limited need for interaction with a classical computer and therefore a limited impact of the hardware and software integration on the algorithm's performance. However, high-performance computing applications typically require large coordination between the different compute nodes, which would benefit from a tight hardware integration. Therefore, it is reasonable to assume that most quantum-classical hybrid workflows for HPC will require significant interaction between the quantum accelerators and the classical computers, as depicted in Figure 2b.

RETHINKING NUMERICAL MODELS

In high performance computing, this implies rethinking of our entire simulation pipeline beginning with the mathematical model. Discretization methods applied previously to a mathematical model will likely not be sufficient or required to implement sub-problems on quantum computers. This challenge presents an opportunity to discover and understand more about our computational field and mathematical methods of solving numerical problems.

Hybrid Classical-Quantum Architectures

The discussion above suggests that the benefits associated to different types of algorithms are highly dependent on the hardware interface between the supercomputer and the quantum accelerator. In this section, we discuss different integration architectures and how they impact the performance of hybrid algorithms, as well as the requirements for quantum-resource management.

HARDWARE: HYBRID SYSTEM PROPERTIES

Two aspects will be of particular focus when considering a shared system design: the locality of the quantum accelerators and their density.

Locality refers to the physical placement of the quantum devices with respect to the HPC system, which can manifest in a wide spectrum, from remote connection to co-located systems sharing a local network to direct on-node connection. Different deployment choices have an impact on data security, operational models, maintenance requirements, financial investment, and communication latency. **Density**, on the other hand, we define by the ratio between the number of available quantum accelerators and the number of classical compute nodes with access to these[3]. Higher densities allow the full hybrid system to be used as a tight interconnected cluster, including the acceleration capabilities, which is the way HPC systems are usually designed and operated. Both locality and density have a direct impact on the performance of different algorithmic classes and need to be jointly considered. In the following, we give some general considerations about these two aspects before looking into their effects on algorithm performance and resource management requirements.

Quantum Accelerator Locality

When considering the hardware interface between the HPC host system and the quantum accelerators, many strategies are possible, including remote, on-premises (or co-located), and on-node integration, as depicted in Figure 3. Each of these brings distinct advantages and disadvantages and will suit different needs.

As the name suggests, in a remote integration, the quantum accelerator is physically distant from the HPC host. In this scenario, quantum computers could be hosted and maintained by the quantum hardware provider, which significantly lowers the adoption barrier. It also allows the supercomputer to gain remote access to several different quantum computing technologies at a lower cost, which can be beneficial, e.g., for pilot quantum projects. However, remote integration presents disadvantages that can be critical to many users. As an example, if the quantum hardware is not hosted by the HPC centre, they will need to rely on data security and integrity provided by a third party as well as by the network connection. For many users, this can be a strong impediment. In a similar approach, for remote connection following standard cloud models, the supercomputer user will share access to the quantum hardware with other cloud users. Cloud models allow pay-as-you-go service, further reducing the investment for users with limited use of quantum acceleration. However, oversubscribed systems will lead to extensive wait-times. For users with high demand for quantum resources, the financial benefits are not as clear, and wait times inherent to shared resources need to be accounted for.

[3] Data asymmetry, the number of qubits compared to the amount of memory available on a classical system is also of concern when integrating the two types of systems, but we consider this an independent topic.

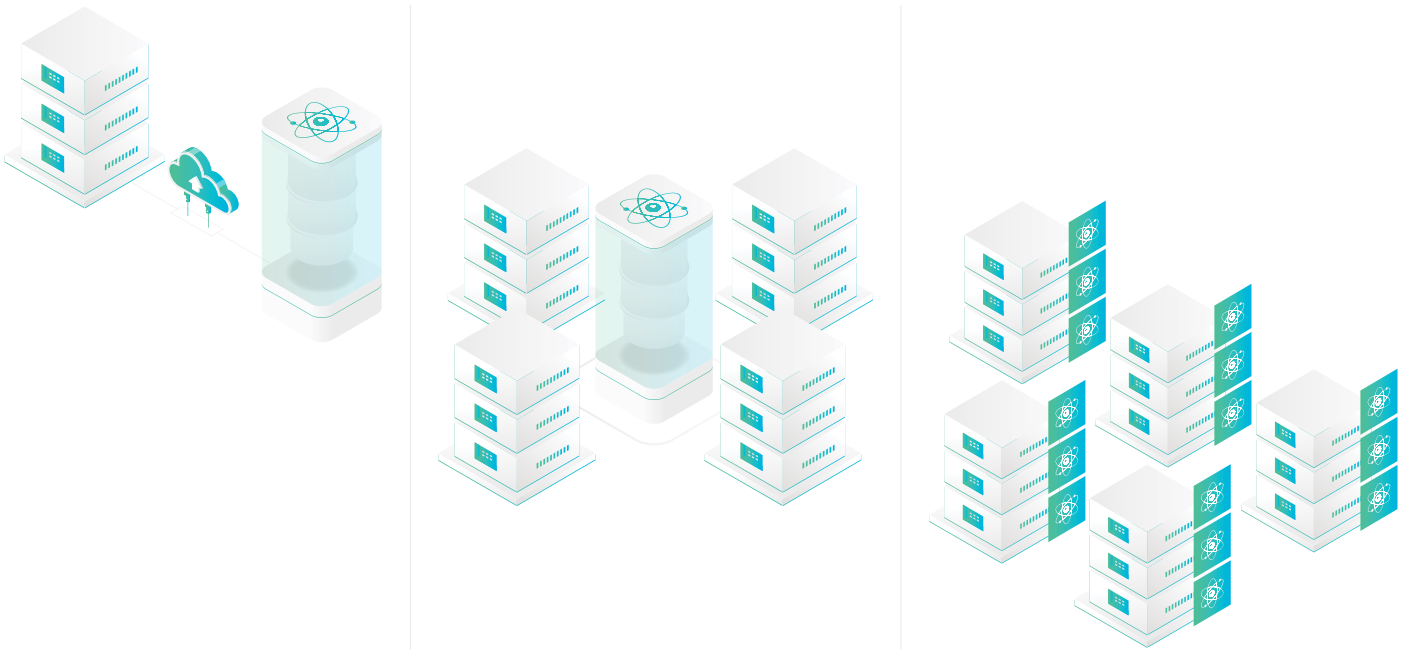


FIGURE 3. DIFFERENT QUANTUM ACCELERATOR DEPLOYMENT MODELS: (left) remote access; (middle) on-premises integration; (right) on-node integration.

For on-premises integration, the quantum hardware is located in close physical proximity to the classical compute infrastructure and connected to the same high-speed interconnect. This immediately solves previous security concerns as the supercomputer host has full control over the quantum accelerators. However, a new security issue arises. As quantum computing technology is still maturing, today's systems require maintenance by specialised staff. For the host entity this implies either allowing the quantum hardware provider to have access to the system or having in-house trained staff. Placing the quantum accelerator in a location where appropriate access can be given may be a viable solution that still profits from the performance and security benefits of on-premises integration. An important advantage of the full control provided by the on-premises approach is the direct system administration, which allows higher control of job scheduling and prioritization according to the host's needs. Very importantly, the reduced latency of on-premises integration allows all different hybrid algorithms to be implemented more efficiently.

The next step in the physical integration is to consider a direct on-node connection to the quantum accelerators. These could range from networking the quantum device directly to the motherboard, to eventually having the CPU and the quantum processor sharing the same die. While on-die integration guarantees the best possible performance, quantum technologies need significant leaps to make this a viable strategy. However, motherboards with specialized sockets for quantum accelerators can have reduced latency, optimized board lanes for the needed bandwidth, and advanced access to shared resources, such as memory. As quantum computing matures, developing this deep integration will allow HPC users to take profit of their full potential.

Quantum Accelerator Density

To understand the role played by the density of quantum accelerators, it is important to consider the near-term scenario as well as expected future developments. Quantum computers hold the promise of significant advances in computational capacities in a wide range of fields. It is then reasonable to expect that many applications will benefit from quantum advantage to some degree. If these many applications need to share a single or small number of quantum accelerators, these resources may become massively oversubscribed.

As an immediate result, this will delay the execution of jobs. For some applications, this delay can be to the point where the time-to-solution would be shorter without using the quantum resource. This points towards supercomputers eventually requiring access to several quantum accelerators, as is the case with other hardware accelerators.

SOFTWARE: HYBRID ALGORITHM PROPERTIES

Alongside the hardware properties, we also need to consider the properties of the algorithm. In particular, here we consider the classical-quantum coupling frequency and the resource usage asymmetry.

Classical-quantum coupling frequency refers to how often the HPC system and the quantum hardware exchange information. While this concept is similar to the concepts of loosely/tightly coupled HPC workloads, coupling frequency refers specifically to the communication frequency with the quantum accelerator. The second concept, **resource usage asymmetry**, refers to the proportion of time the computation uses HPC or quantum hardware. For applications with large imbalance, one of the resources will be idle for a significant time and unnecessarily add to the wait times for subsequent jobs, as well as to the financial costs.

To understand the impact of these, let us consider a few specific cases illustrated in Figure 4. For remote integration, the unavoidable latency added by the long-distance network communication, as well as the necessary security protocols, plays an important role in the performance, or even feasibility, of different algorithms. Consider a hybrid algorithm that offloads a small number of tasks to the quantum accelerator. The increase in the algorithm's time-to-solution coming from network communication has minimal impact on the overall system's performance. Typically, the more coordination is needed between classical and quantum compute nodes and the higher the frequency of communication between all compute nodes, the more impactful will be the effects of the added latency.

It should be noted, besides increased time-to-solution, another consequence for the performance of the full compute system is that the compute nodes remain idle during the communication cycles, i.e., latency leads to unnecessary waste of clock cycles. If hundreds or even several tens of thousands compute nodes are waiting for the input of the quantum accelerator, these lost clock cycles can be costly to the HPC host and its user. Importantly, this will also be detrimental to the energy-to-solution and compromise the sustainability provided by the quantum accelerator.

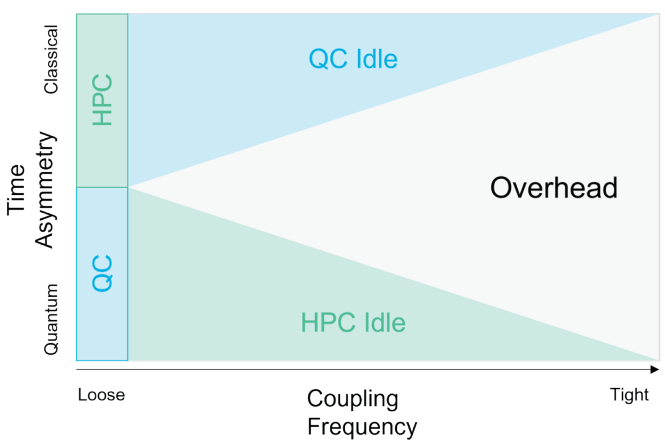


FIGURE 3.

Region plot within which any classic-quantum application can be placed. The regions show important time fractions of the application and resource usage. A purely classical or quantum application is placed on the left-hand side, while an evenly balanced hybrid application is placed along the horizontal centre line. As the coupling frequency increases the higher the time fraction of the overhead becomes.

In extreme cases, long-distance network latency is not only costly but renders the result of the entire calculation unusable. This is the case for algorithms that require the input of a classical computation to happen within the lifetime of the qubits. A concrete example is the optimal decoding of stabilizer codes, which is part of standard protocols for quantum error correction. This is an #P-Complete problem [10], which can only be offloaded to the HPC system if the total communication and classical processing times can happen sufficiently under the qubits lifetime.

Notice that for applications with large imbalance and low coupling frequency, the system's performance can be boosted by the ability to independently manage HPC and quantum resources. However, any such solution will have increasing complexity as the coupling frequency increases.

BENEFITING FROM ON-PREMISES INTEGRATION

Following the arguments above, for on-premises integration, it is useful to consider two common HPC workloads, loosely and tightly coupled workloads. In the former, a user submits a very large number of tasks that can be run in parallel, and which require little or no communication between them. Typically, the number of tasks is significantly larger than the number of compute nodes used, so in practice, once the maximum parallelism is achieved, the following tasks run sequentially. If these tasks offload work to the quantum accelerators, the total communication latency can add a significant delay to the time-to-solution of the algorithm. These applications typically use a large number of nodes, so even in cases where the number of sequential tasks is not significant, the total computation time lost due to network latency can be significant, again impacting the costs incurred by the user and the overall system's performance.

In tightly coupled workloads, the computation consists of small tasks that can run in parallel, but that require input from one another, therefore message passing between the compute nodes occurs at high frequency. As before, communication latency may substantially increase the time-to-solution depending on the details of the workload and the level of classical-quantum coupling frequency. Even more so than in other cases, as a high level of coordination is needed, several compute nodes might depend on the results from a single quantum accelerator and can therefore remain idle. These scenarios showcase the performance advantages of quantum accelerators in the same network fabric as their HPC host system.

Different architectural choices and their technical implications will affect differently the performance of the various algorithms. Initially, user awareness of this impact will be paramount in selecting the best resources for a given application. However, the development of a resources management system tailored to the optimal use of quantum accelerators is central to improving the user experience and speedup user adoption.

System-wide Quantum Resource Management

To facilitate an on-premises hybrid classical-quantum supercomputer, resource allocation needs to be managed carefully. This is a similar problem solved by batch schedulers used in compute centres at present. A resource manager aims to maximize the usage of resources between users of a system and prevent conflicting access between users. SLURM [18] is the most commonly used batch scheduler today, which handles the exclusive allocation of compute resources to the submitted jobs. These resources can be, e.g., compute nodes, cores, memory, or GPUs.

Large compute resources are typically shared, both spatially and temporally, across the users. Rarely will a single project utilize the entirety of a compute resource. Historically, this is a common practice; early computing machines were time-shared. Subdivision of hardware spatially, allocating nodes to compute jobs, has its origin from the time when supercomputers became clusters of nodes compared to specialized monolithic computers. To handle this subdivision, a batch scheduler is used as a top-level entity with which users interact and submit *jobs* to.

The scheduling algorithms employed were developed to satisfy all the user requirements, e.g., IO, GPU usage, memory usage, and priorities, and execute all possible jobs as soon as possible. The goal is to have the least number of resources idle at any moment, thus maximizing the economic return of the investment. Therefore, the scheduler needs to determine the heuristically best possible ordering of all jobs spread across all resources. Evaluating the best schedule is computationally ineffective. In addition to the allocation size, projects typically have priority levels, which introduce another dimension of complexity in scheduling. The usual architecture for modern day supercomputers is to have as few contented resources as possible, this is seen by, e.g., equipping every node with GPUs instead of a small subset. If these resources would have to be shared non-exclusively, then resource starvation and task convoying would occur frequently. Resource starvation is the situation in which a task is perpetually blocked from making progress, as the necessary resources are never made available. Task convoying is seen when a long running task occupies a resource, while shorter tasks are unable to access this resource to also make progress.

Quantum computing integrated with HPC causes the scheduling to be more constrained as quantum computers have intrinsic limitations that classical hardware does not have, such as the exclusivity of resources or the bounded execution time of quantum algorithms in current systems. In respect to scheduling, quantum computing introduces the indivisibility of a single shot (a sampling of the quantum state which is generated using a quantum circuit) and is also intrinsically not pre-emptiable, as the state cannot be saved, unlike in a classical computer.

SLURM is flexible, configurable, and open source, which makes it ideal from a system administrator and research perspective. However, SLURM was designed for batch scheduling of classical resources with typically exclusive access to a node and all its associated resources. As seen in Figure 5, the time range which it governs is limited to the high end from minutes to days. This can be seen in the minimal tick frequency able to be set: once per second. The tick frequency is the rate at which schedule recalculation happens if a scheduling affecting event occurs. The minimum tick frequency of once per second is good enough for all use-cases a batch scheduler would encounter in a compute centre.

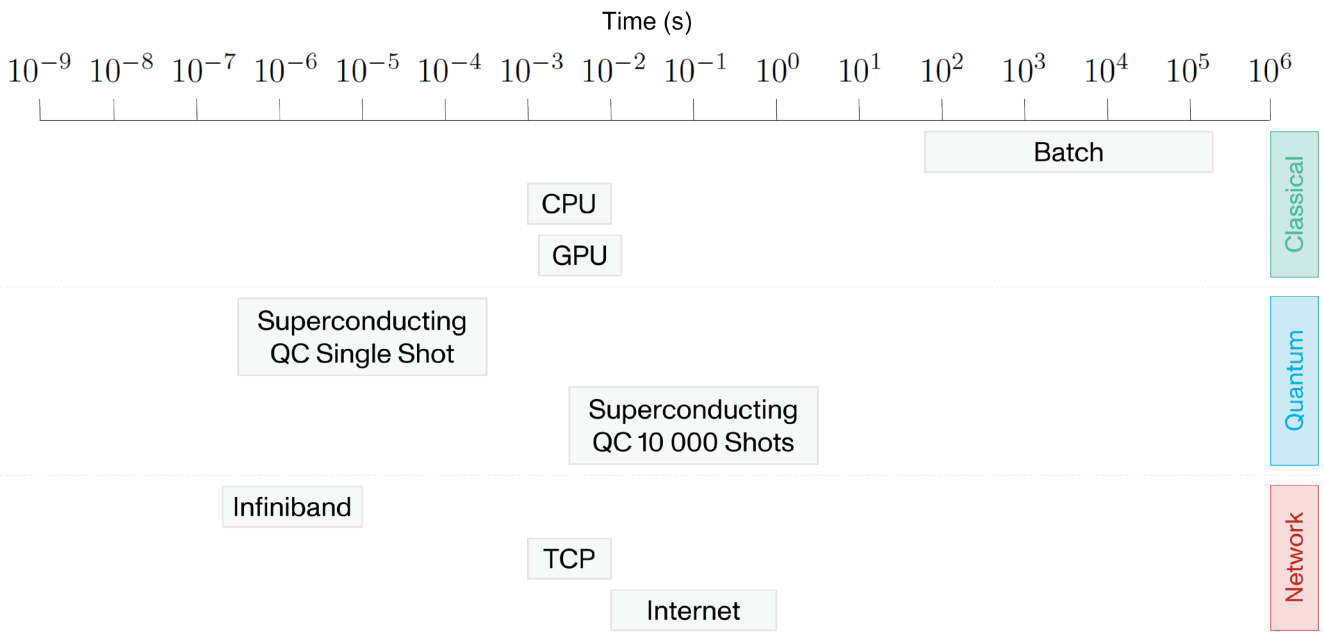


FIGURE 5. *Time ranges for which resource managers or scheduler are usually responsible. As seen quantum combines the batch scheduling ranges with very small time scales as shots. This is a unique combination for which a batch scheduler is not equipped to handle. The time ranges presented are estimates for each category.*

Since SLURM is open source, it could be modified to allow for a higher frequency, but likely this will require significant reworking of the entire scheduler, since multiple tenancy is another requirement for a quantum scheduler. Researchers also avoid modifying SLURM and instead choose to modify more modern schedulers, like Flux [1] or OAR [3], due to the lower complexity. Due to this large work requirement, an independent development of a system-wide quantum task scheduler is preferable, with the design and implementation able to focus specifically on the quantum workloads and their complexities, instead of spending development resources on covering the entire timescale shown in Figure 5. Figure 6 shows the access patterns which would be supported using a system-wide resource manager in a compute centre. The focus remains to support all forms of usage of a quantum computer, while also enabling access from the HPC systems to facilitate larger scale computations.

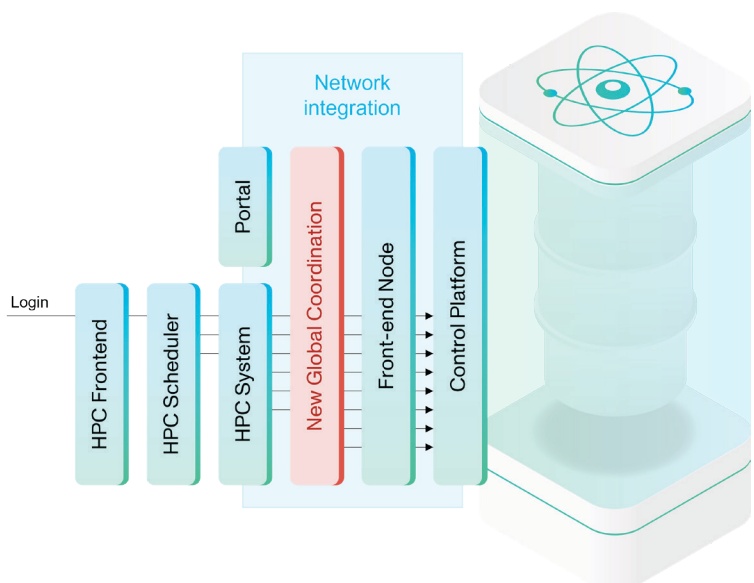


FIGURE 6. MECHANISMS OF ACCESS TO QUANTUM RESOURCES PRESENT IN A HYBRID CLASSICAL–QUANTUM SUPERCOMPUTER.

Like past developments of batch schedulers, it is natural for compute centres to take a central role in developing such a system-wide resource manager as they are key stakeholders. In addition, and as discussed above, compute centres have security interests which supersede even performance concerns. The on-premises concept for future hybrid classical-quantum computers is required for this, since transmitting confidential information to remote services (even in abstract forms) would be disallowed by policies. Delegating this task to users would solve the security questions on a per-case basis, but would cause widespread misalignment of performance, quality of implementation, and duplicated development effort. To reduce fragmentation, it is central to the compute centre's role to provide this resource management directly to its users.

OFFLOADING

Given a large asymmetry between classical and quantum resources, the early implementations that truly benefit are those which use the quantum resource sparingly but still facilitate a large reduction in runtime.

To allow this within a supercomputing environment, a scheduler needs to allow shared access to the quantum computer. Exclusivity would potentially cause task starvation or task convoying, which in effect causes the classical resources to remain idle. Interleaving of multiple statistical samples from different jobs is likely to be used as a path forward while the quantum resources are constrained. Interleaving of shots allows all tasks to be executed in a fair way allowing more shots for higher priority tasks. This method prevents starvation of a classical component of a job by ensuring equal priority jobs get equal treatment. The exact parameters for interleaving are tuneable and therefore allows system-wide administration of access to quantum resources. One down-side of interleaving is that current control electronics require a significant amount of time to reconfigure. Given the specific parameters, this base time can be too large to allow for any effective swapping between workloads, and effectively serializes all workloads across the classical-quantum boundary.

At the beginning of the distributed hybrid classical-quantum applications era, bursty behaviour is expected, with many quantum tasks spawned in short order. One workflow leading to bursty behaviour would be from bulk synchronous parallel [17] applications from HPC, which submit quantum tasks in almost unison. This burst of tasks will need to be processed on all available quantum resources as soon as possible to enable progress on the classical resources. This is disadvantageous since it causes large delays; if few quantum resources are available and are flooded; the quantum resources will be oversubscribed.

These artefacts are also observed in other bursty systems and are fundamentally unavoidable. The only way to properly address this issue is by having enough quantum resources present at any time to be able to handle large spikes in demand. On the other hand, due to the blocking nature and dependency of the workloads, we could see a self-adjusting mechanism which would mitigate these effects to some degree between multiple applications.

RESOURCE SCHEDULING

When using multiple quantum resources, other challenges arise as well. At least in the early generations, quantum processing units will be notably variable in terms of stability and capabilities. In a sense, they would form a heterogeneous computing infrastructure themselves. In addition, the underlying technology may render some quantum processing units more suitable for certain tasks than other technologies. This is a significant departure from current classical systems, where (roughly) identical resources can be manufactured and purchased. Therefore, the scheduling choice also matters in terms of which specific unique quantum processor the task is delegated to, which strengthens the argument for a specialized system-wide resource manager for the quantum accelerators.

While workloads will target abstract quantum processing models, there may be hardware-specific workloads which will require a specific resource [6]. These concerns for quantum computing would be: qubit count, coherence time limits, native gate sets, gate fidelities, and qubit connectivity, in addition to others. Native gate sets are gates that can be directly implemented in the hardware system without the need for transpilation. These would act as a soft limit in this situation, since a universal gate set can emulate any other gate set. But a closer match of the algorithms operations to the native operations of the quantum system could increase the quality of the computation. Hard constraints such as the qubit count are unavoidable. To address this, general constraint solving mechanisms can be applied, but additional onload work will be required.

ONLOADING

While the goal with quantum acceleration is offloading a task to ensure reduced time and energy-to-solution, these systems can benefit from onloading work, which consists of work done to facilitate the quantum computation but does not directly progress the computation of a result. In classical computing, the synthesis and compilation of a program are done at compile-time, which is a fixed time prior to any runtime and therefore is an amortized cost. Due to the general requirement for a quantum circuit to be synthesised and compiled or optimized at runtime, the onload concept becomes relevant. This encompasses quantum circuit synthesis, compilation, optimization, error detection, system calibration, and many other aspects.

The frequency at which these computational tasks must be performed varies. Compilation and optimization would have to be performed when switching quantum resource or circuit. System calibration is performed when it is noticed that the hardware no longer produces the correct results. Error detection and correction will be required throughout the process of executing a single quantum circuit.

While these onload workloads must be completed to enable a result to be gathered, it is typically not desired to dedicate a large number of exclusive resources directly to this task; this would result in idle resources which could be used for offloading workloads instead. For this, a likely solution would be to use idle classical resources waiting for quantum task execution to be utilized to perform onload work, increasing the overall efficiency across the entire system.

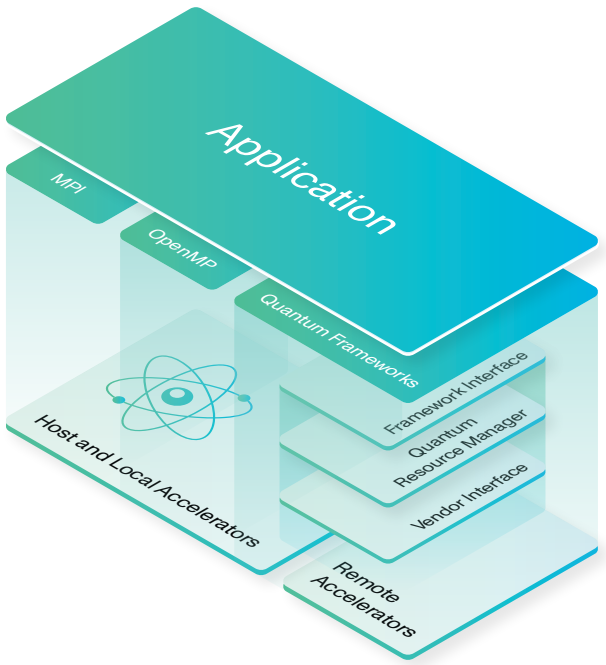


FIGURE 7: SOFTWARE STACK DIAGRAM FROM THE QUANTUM INTEGRATED HPC APPLICATION TO THE QUANTUM HARDWARE.

Interfaces are explicitly indicated which must be standardized to allow for vendor or framework interoperability.

CONTROL ABSTRACTION LAYERS

We anticipate future hybrid classical-quantum supercomputers to have multiple quantum accelerators attached at various levels of the system hierarchy, similar to classical compute units and memory. This forces a standardization of interfaces between the quantum computer software and the system-wide resource manager to enable abstraction from the specific resource a user is occupying. Figure 7 shows the abstract software stack required in addition to the two interfaces required to facilitate compatibility to multiple vendors and frameworks.

The *framework interface* allows any top-level quantum framework or programming language runtime to access all quantum resources available. Frameworks such as Cirq [5], Qiskit [2], XACC [13], would require a plugin to be written for them to interact with such an interface. This allows any future framework to target the system-wide resource manager. This reduces the barrier for users to engage with the technology, thus improving user experience and accelerating widespread adoption. The key difference is that optimization of quantum programs is moved below this layer and only simple offload control is propagated upwards.

The vendor resource interface would expose several categories of control of the quantum hardware. First, the vendor APIs must support typical HPC management functions, such as submission, querying, modification, and cancelling of a quantum program task. This enables basic control orchestration from the system-wide resource manager. In addition to the control category, a query interface needs to be present to allow introspection of the quantum resource for the resource manager to decide which quantum program should be executed on which quantum resource.

Current vendor implementations prioritize ease of use and development speed over performance, typically with a set of HTTP APIs or using Python. These implementations would need to be revisited in the context of a high-performance system and potentially mandated to be more performance focused. Finally, while offloading quantum tasks is supported by the above requirements, onloading of tasks from the quantum resource must also be enabled. In other words, the quantum accelerator interface needs to provide execution capabilities for the reverse path to allow onload work to be delegated from a quantum control node to the classical resource nodes.

By addressing the requirements discussed above, the resource manager removes obstacles for adoption of this upcoming technology for both the users and the HPC centre, while at the same time being a step towards optimizing the usage of quantum accelerators.

Outlook

Integrating quantum accelerators into high performance computing systems is key to reach the full potential of both systems. These upcoming hybrid systems will allow the large community working on HPC applications to significantly increase the boundary of what can be computationally investigated, and it will also invite new domains into the field of high-performance computing. To achieve this, several aspects of this integration remain open, both in the software and hardware sides.

In a community known for its legacy software usage, a complete overhaul of existing application code is unlikely to occur. Software written over decades by the HPC community will have to be carefully analysed to determine a path allowing it to take advantage of quantum computing. In the future, the community will likely integrate quantum kernels in a similar manner to XACC or CUDA, or potentially using offload or compute interfaces such as OpenMP. The primary concern would be the friction of adoption of any individual expression of such an integration. In this regard, it is critical that the quantum and HPC stakeholders align towards standardization, which will accelerate the technological development and its adoption.

As is the case for current high-performance computing, different deployment models of quantum accelerators will enable different use-cases, with on-premises installations providing the best performance and security with the current technology. Quantum hardware manufacturers and compute centres must jointly develop this integration, in order to ensure a long-term match and sustainable growth of quantum computing in an existing ecosystem.

It is imperative that the development of this integration happens now, as the challenges for this integration are manyfold and we need to pave the road as quantum systems continue to grow. In order to optimize return on investment, the necessary strategies for the integration must be in place once the quantum hardware is ready to provide commercial and scientific advantage. Importantly, deploying the first generations of quantum accelerators now will speed-up the development of new hybrid algorithms by the HPC user community. The good news is that several initiatives already exist in that direction and the first on-premises quantum accelerators will be delivered in 2023 [7]. These are the first steps in the road ahead to build the future of computing and to help build solutions for the well-being of humankind.

Acknowledgements

The project on which this report is based was funded by the German Federal Ministry of for Education and Research under the grant number 13N15689 (DAQC), 13N16063 (Q-Exa), and the Bavarian State Ministry of Science and the Arts as part of Munich Quantum Valley.

The responsibility for the content of this publication lies with the authors.

References

[1] Dong H. Ahn, Jim Garlick, Mark Grondona et al: A next-generation resource management framework for large HPC centers. In *2014 43rd International Conference on Parallel Processing Workshops*, 2014.

[2] MD SAJID ANIS, Abby-Mitchell, H´ector Abraham et al. Qiskit: An open-source framework for quantum computing, 2021.

[3] N. Capit, G. Da Costa, Y. Georgiou et al. A batch scheduler with high level components. In CCGrid 2005. *IEEE International Symposium on Cluster Computing and the Grid, 2005.*., volume 2, pages 776–783 Vol. 2, 2005.

[4] M. Cerezo, Andrew Arrasmith, Ryan Babbush et al. Variational quantum algorithms. *Nature Reviews Physics*, 3:625–644, sep 2021.

[5] Cirq Developers. Cirq, August 2021. See full list of authors on Github: https://github.com/quantumlib/Cirq/graphs/contributors.

[6] Bundesministerium für Bildung und Forschung. Digital-analoge quantencomputer, 2021.

[7] Bundesministerium für Bildung und Forschung. Quantencomputer-erweiterung durch exa-scale-hpc, 2021.

[8] Secrétariat général pour l'investissemen. Stratégie quantique: lancement d'une plateforme nationale de calcul quantique, 2022.

[9] IDC. Untangling the HPC Innovation Dilemma through Quantum Computing, November 2021.

[10] Pavithran Iyer and David Poulin. Hardness of decoding quantum stabilizer codes. *IEEE Transactions on Information Theory*, 61(9):5209–5223, 2015.

[11] Stephen Jordan. Quantum algorithm zoo, 2021.

[12] Top500 The List. November 2021 — top500, 2021.

[13] Alexander J McCaskey, Dmitry I Lyakh, Eugene F Dumitrescu et al. XACC: a system-level software infrastructure for heterogeneous quantum–classical computing. *Quantum Science and Technology*, 5(2):024002, feb 2020.

[14] Michael A. Nielsen and Isaac L. Chuang. *Quantum Computation and Quantum Information: 10th Anniversary Edition*. Cambridge University Press, 2010.

[15] High Performance Computer – Quantum Simulator. High Performance Computer – Quantum Simulator, 2021.

[16] EuroHPC Joint Undertaking. Call for expression of interest for the hosting and operation of european quantum computers integrated in HPC supercomputer, 2022.

[17] Leslie G. Valiant. A bridging model for parallel computation. *Commun. ACM*, 33(8):103–111, aug 1990.

[18] Andy B. Yoo, Morris A. Jette, and Mark Grondona. Slurm: Simple linux utility for resource manage-ment. In *Job Scheduling Strategies for Parallel Processing*, 2003.

Contact us

Are you interested to hear more?
Reach out to us.

ABOUT IQM QUANTUM COMPUTERS

IQM is the Pan-European category leader in building quantum computers.

IQM delivers on-premises quantum computers for supercomputing centers and research laboratories and provides complete access to its hardware. For industrial customers, IQM delivers quantum advantage through a unique application-specific co-design approach. IQM is building Finland's first commercial 54-qubit quantum computer with VTT, and an IQM-led consortium is building Germany's quantum computing system that will be integrated into an HPC supercomputer to create an accelerator for future scientific research.

IQM is also part of Atos Scaler program.

For more information, visit: www.meetiqm.com.

Jani Heikkinen
Head of Business Development,
Country Manager, Spain

jani.heikkinen@meetiqm.com
+358403642706

Bruno G. Taketani
Team Lead, HPC Integration

bruno.taketani@meetiqm.com
+49 171 550 4155

ABOUT LEIBNIZ SUPERCOMPUTING CENTRE

The Leibniz Supercomputing Centre (LRZ) proudly stands at the forefront of its field as a world-class IT service and computing user facility serving Munich's top universities as well as research institutions in Bavaria, Germany and Europe. As an institute of the Bavarian Academy of Sciences and Humanities, LRZ has provided a robust, holistic IT infrastructure for its users throughout the scientific community for nearly sixty years. It offers the complete range of resources, services, consulting and support – from email, web servers and Internet access to virtual machines, cloud solutions, data storage and the Munich Scientific Network (MWN). Home to SuperMUC-NG, LRZ is part of Germany's Gauss Centre for Supercomputing (GCS) and serves as part of the nation's backbone for the advanced research and discovery possible through high-performance computing (HPC). In addition to current systems, LRZ's Future Computing Group focuses on the evaluation of emerging Exascale-class architectures and technologies, development of highly scalable machine learning and artificial intelligence applications, and system integration of quantum acceleration with supercomputing systems.

For more information, visit: https://www.lrz.de/

Sabrina Schulte
Head Of Public Relations

presse@lrz.de

Martin Ruefenacht
Research Scientist, HPCQC Integration

martin.ruefenacht@lrz.de
+49 89 35831-7863