Will Quantum Computers Scale without Inter-Chip Comms? A Structured Design Exploration to the Monolithic vs Distributed Architectures Quest

Rodrigo, Santiago; Abadal, Sergi; Alarcon, Eduard; Almudever, Carmen G.

DOI
10.1109/DCIS51330.2020.9268630

Publication date
2020

Document Version
Final published version

Published in
2020 35th Conference on Design of Circuits and Integrated Systems, DCIS 2020

Will Quantum Computers Scale Without Inter-Chip Comms? A Structured Design Exploration to the Monolithic vs Distributed Architectures Quest

Santiago Rodrigo, Sergi Abadal, Eduard Alarcón

NaNoNetworking Center in Catalonia
Universitat Politècnica de Catalunya
{srodrigo, abadal}@ac.upc.edu, eduard.alarcon@upc.edu

Carmen G. Almudever

Quantum and Computer Engineering Dept.
and QuTech
Delft University of Technology
C.GarciaAlmudever-1@tudelft.nl

***Abstract*—Being a very promising technology, with impressive advances in the recent years, it is still unclear how quantum computing will scale to satisfy the requirements of its most powerful applications. Although continued progress in the fabrication and control of qubits is required, quantum computing scalability will depend as well on a comprehensive architectural design considering a distributed multi-core approach as an alternative to the traditional monolithic version, hence including a communications perspective. However, this goes beyond introducing mere interconnects. Rather, it implies consolidating the full communications stack in the quantum computer structure. In this paper, we propose a double full-stack architecture encompassing quantum computation and quantum communications, which we use to address the monolithic *versus* distributed question with a structured design methodology. For that, we revisit the different quantum computing layers to capture and model their essence by highlighting the open design variables and performance metrics. Using behavioral models and actual measurements from existing quantum computers, the results of simulations suggest that multi-core architectures may effectively unleash the full quantum computer potential.**

I. Introduction

Research on quantum computation forged since the 80s has created considerable expectations on its unprecedented processing power and unconditional security, which could change forever key areas such as cryptography, big data analysis, AI, and biochemistry (drug synthesis) [1]–[4]. By leveraging quantum mechanical properties such as superposition and entanglement, quantum computer implementations of time-consuming algorithms can be exponentially faster than their classical counterparts. [5].

However, preserving these key properties implies maintaining the quantum information in qubits (the *alter ego* of classical bits in the quantum world) intact, i.e. keeping information coherence. This, being trivial in classical computing, is in fact one of the most challenging issues for building quantum computers: quantum processors must be kept at very low temperatures (close to the absolute zero) and isolated from the outside world, something which makes the external control and computation, for operations on the qubits and measurements of their values, a very challenging task.

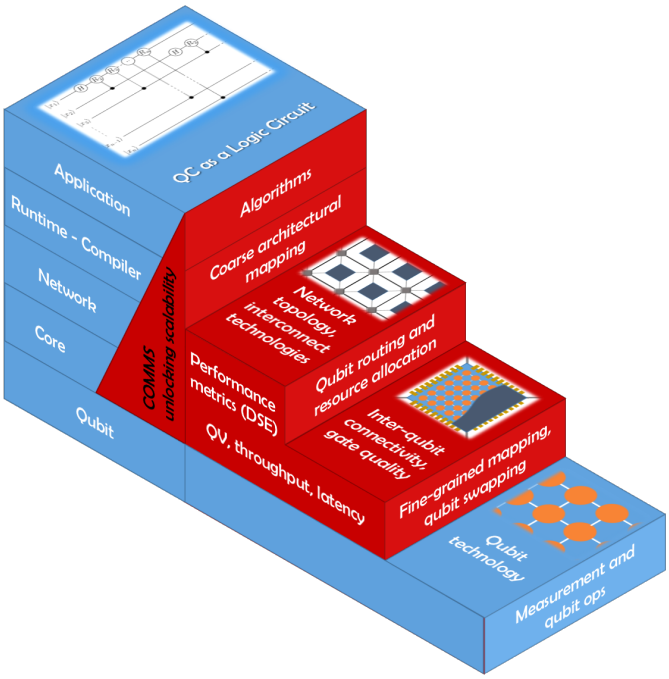


Fig. 1. Double joint full-stack layered architecture for multi-core quantum computers

Although during these last years we have seen remarkable sustained advances on quality and number of qubits in working prototypes, the existing realizations of quantum computers are too small-scale and error-prone yet to be able to experimentally demonstrate the theoretical results and proven algorithms that show these impressive speed-ups. In fact, current approach for designing and building quantum computers, based on densely integrating qubits on a single chip, is conjectured not to scale past some hundreds of qubits, due to impracticality

978-1-7281-9132-4/20/$31.00 ©2020 IEEE

of control circuits integration, per-qubit wiring, prohibitive quantum decoherence and severe qubit operation errors [6].

We postulate that, even though the challenges are hard and diverse, a comprehensive approach of the computer design based on multi-core architectures, as opposed to current densely-packed monolithic approach, is crucial to unlock the scalability issues. This multi-core quantum computer, will cluster together dozens of Noisy Intermmediate-Scale Quantum (NISQ) computers[1] cores (with tens to hundreds of qubits), connected through a quantum communications network (for core-to-core qubit transport, such as quantum teleportation or photonic switches) and a control classical network (for core coordination and job distribution), mapping the quantum algorithm among them to boost performance: in this way, we alleviate the requirements for control circuits and improve qubit isolation, while leveraging all the advantages of quantum parallelism.

Various proposals [8]–[14] in the existing literature agree on using this approach. Existing articles use different qubit technologies (ion trap, quantum dots or impurities in solids) and *module* interconnects (ion shuttling, photonic switches, quantum teleportation), but to the best of our knowledge none of them has deeply analyzed whether this multi-core approach is effectively enabling an architecturally scalable quantum computer, and which are the resource overheads and computational costs of such architectures.

Therefore, we aim at substantiating that quantum computing scalability (and ultimately, quantum computing culmination) may not be possible without multi-core architectures enabled by communications, in a Quantum Network on Chip (QNoC) fashion. We do so by performing a first analysis of this approach focusing on the intermodule (or interchip) communication.

In this article, we set the framework for this analysis: we propose a double full-stack layered architecture combining communications with traditional quantum computer designs, and present a Design Space Exploration (DSE) formulation to this problem. DSE will be used to compare the multi-core and monolithic single-core approaches and find a sweet spot (or region of the design space) where the design performs better.

II. Connecting the (quantum) dots

In this section, a generic multi-core quantum computer architecture is presented in order to facilitate the context for the analysis previous to the DSE, which will identify the parameters and performance metrics that best fit our problem.

Some layered architectures for quantum computing have already been proposed [15]–[17], but all of them focus on single-core quantum computers, lacking for a communications perspective. We introduce a generic (i.e no specific qubit technology or interconnect technology is assumed) layered stack that implies multi-core quantum computing by adding the corresponding layers and identifying the communication processes that may be involved. This approach goes beyond adding mere interconnects, encompassing instead communications and computing in a consolidated layered architecture itself *–a là* NoC (Network-on-Chip) [18]–. Although there exist some stack proposals extending quantum computers to connected environments, these approaches come from a Quantum Internet perspective, i.e. do not integrate the quantum computation process with communications: they are network stacks rather than computer architecture stacks [3], [19], [20].

The full-stack layered architecture for multi-core quantum computers proposed in this paper can be seen in Fig. 1. In order to represent the different abstractions of the quantum computer at each of the layers, we have included a *stairway* that graphically explains what elements configure that specific layer (on each of the step treads) and its key functions (on the step risers).

Application Layer. The upper-most layer is composed by the code of the quantum application/algorithm to be run on the quantum computer. In this layer the quantum computer is seen as a Logic Circuit with no reference to limits and architecture for communication among qubits. Nonetheless, the code could include some compiler instructions enabling optimized qubit distribution and instructions execution, as it is already done in multi-core classical computing.

Runtime/Compiler Layer. It is in charge of translating the human-written code to a machine-adapted assembler code (compilation) and coordinate the instructions execution and the coarse architectural mapping (i.e. partitioning of the algorithm among the existing cores, in analogy with the *mapping* process in classical many-core computer architectures), always in pursuit of an optimized processing. Therefore, it needs a closer look to the architecture, knowing about the capabilities and topology of the multi-core quantum computer. However, it is not directly in charge of communicating qubits.

Network layer. This layer (which forms what we could call a Quantum NoC, i.e. a QNoC) may receive some instructions from the compiler regarding qubit movement among cores, but it is fully responsible of selecting the best time and route to do so, as well as to optimize all the inter-core[2] communication by reserving resources or preparing qubit movements in advance. It might implement different inter-core topologies (such as all-to-all, star, ring or regular 2D lattices) as well as interconnect technologies (e.g. ion shuttling, qubit teleportation...). Both compiler and network layers *see* the quantum computer as a set of quantum cores (i.e. "processing units") connected in a certain topology. Communications are crucial at these layers, as they are ubiquitous in every action performed at this level.

Core layer. In a single-core quantum computer (no network layer), this one represents in fact the whole computer. In any case, the core layer's view is reduced to a set of qubits integrated in a single core capable of interoperate using one and two-qubit gates. It performs the fine-grained qubit mapping inside the core as well as inter-core I/O operations

[1]NISQ is a term coined by John Preskill that encompasses the small-sized and constrained (yet fascinating) computers built nowadays [7]

[2]*Inter-core* communication involves transferring qubits *among* cores, while *intra-core* communication refers to any type of qubit transmission happening *inside* a core.

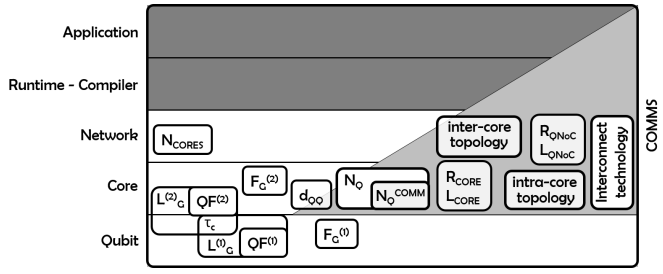


Fig. 2. Parameter space, placed within their respective layers in the stack

control. Therefore, communications play also here a remarkable role, as qubit swapping is the most basic form of quantum communication, and the core needs also to receive input values and send results to other cores. Qubit connectivity inside a core (encompassing topology and communication type), as well as gate quality are key elements configuring this layer.

Qubit layer. The last one is the qubit layer: the individual qubits, whether they are logical qubits (i.e. a group of qubits acting as a single qubit in order to reduce operations error and increase coherence) or physical (Quantum Error Correction (QEC) techniques should be applied instead to handle limited fidelity). No communications are involved, but being the foundation of the whole computer, the performance of this layer is key, as is further explained in the next section. Decoherence processes as well as measurement and gate performance are the main aspects here, and at the same time are highly dependent upon the qubit technology.

This full stack overview of a multi-core quantum computer with built-in communications helps us to show that they play a fundamental role not only in a specific part of it, but in the computer as a whole: without the communications block (in red), the stack in Fig. 1 is unstable. But, the question arises of whether this key block would really unlock quantum computer scalability.

III. Compressing a quantum computer through models: distilling its essence

In order to adequately select the crucial elements in the design for the optimization problem we are facing –whether a multi-core architecture can make the difference in terms of processing power scalability–, we must particularly take into account the elements of the quantum computer that may be affected by the architecture paradigm (single-core *versus* multi-core). For this reason, basing upon the layered stack described in the previous section, we will analyze the three lower layers that come into play when considering multi-core architectures, and thus are most affected by intra-core and inter-core quantum communications, namely: qubit, core and network layers (see Fig. 2).

A. Qubit layer

When looking at an individual qubit for the main features that may affect the performance of the quantum computer as a whole, the analysis must take into account the different available technologies (ion traps, superconducting qubits, quantum dots, NV centers in diamond...), as there is no dominating qubit technology yet. Each technology is on a different maturity stage, and presents advantages and disadvantages on the various qubit quality attributes.

Three main parameters describe the performance of a qubit. First, the coherence time (τ_c) sets a fundamental limit on the maximum time we can operate and read out the state of the qubit. Short coherence times mean short-lived variables –which in turn implies that complex algorithms are not supported–. Following the literature, we take τ_c as the empiric value T_2 (phase damping). Because of space restrictions, we will not enter into details in the present paper.

Second, the quality factor (QF) is a parameter derived from the coherence time τ_c and the gate latency L_G (the time spent in performing a certain quantum operation, such as a Hadamard gate or a CNOT): it is an estimate of the number of gates (quantum operations) that can be applied to a qubit while it contains coherent information. The Q-factor is computed as $QF = \tau_c / L_G$.

And finally, gate fidelity (F_G), which is a simplification of the complex quantum error models, and represents how likely a quantum operation will not introduce errors in the system. Low fidelity values will render useless a qubit, no matter how long the coherence time might be.

Effect on communications. The qubit layer is not directly related to any communication process, as we have stated in the previous section. However, being the foundation of the whole computer, it will impose some limits on latencies and throughputs of upper layers communication processes. This is particularly relevant when we consider that quantum communication is closer to "transporting physical qubits" rather than to "sending quantum information". The effects of these parameters on the upper layer communication processes are indirect but real: e.g., a short coherence time might cause a qubit state communication to completely fail, if the time-of-travel is longer than τ_c. Long gate latencies (i.e. small quality factors) have a similar effect: qubits supporting long travels will not withstand too many operations on its already worn quantum state. Finally, low fidelities are equivalent to the inverse of classical communications error rates.

B. Core layer

In the previous subsection we have introduced τ_c, QF and F_G. Although the coherence time is directly related to a single qubit, the quality factor and the gate fidelity are usually computed separately for one-qubit gates ($QF^{(1)}, F_G^{(1)}$) and two-qubit gates ($QF^{(2)}, F_G^{(2)}$). Therefore, from the layered stack perspective, *two-qubit fidelity* $F_G^{(2)}$ and *two-qubit quality factor* $QF^{(2)}$ would in fact become the first parameters of the core layer, as they involve operations among more than one qubit.

We will use N_Q^{CORE} for the *total number of qubits* forming the core. If they are integrated in a multi-core architecture, N_Q^{COMM} of them will be responsible for interconnecting the core with one or several (identical) modules. In the extreme case of a monolithic single-core architecture, no other module

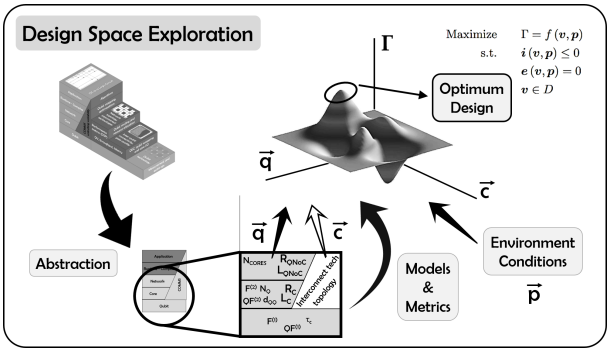


Fig. 3. A Design Space Exploration for Multi-Core Quantum Computers

exists and hence $N_Q^{COMM} = 0$. The interconnection graph might follow a certain topology, whether it is all-to-all, a ring or a regular 2D lattice. Together with the inter-qubit communication technology, it characterizes the *intra-core connectivity*. Finally, the control wiring and qubit technology determine a global minimum *qubit-to-qubit distance* d_{QQ} across the system, which will limit the area occupied by the core and affect the communication latencies.

Effect on communications. Inside a quantum core, the most common form of communication is direct swapping, which involves a series of SWAP gates to move the quantum state from any qubit to another one in the same core. The performance of this communication process will be clearly affected by low values of $F_G^{(2)}$ and $QF^{(2)}$, as well as by the topology and the number of qubits N_Q^{CORE}: a large processor with an uneven topology may need on average longer travels. In other types of communication, such as qubit shuttling, the inter-qubit spacing will determine the travel distance (and duration). In any case, it is of interest for the analysis to derive a mean *intra-core communication latency* $\bar{L}_{CORE}$ *and throughput* $\bar{R}_{CORE}$.

C. Network layer

At this layer, we can see the whole quantum communications (QNoC) perspective. Parameters interesting to our analysis include *inter-core connectivity* (both in terms of topology and interconnect technology) and the *number of cores* in the processor N_{CORES}.

Effect on communications. Depending on these parameters we will obtain different values for mean *inter-core communication latencies and qubit rates* ($\bar{L}_{QNoC}$ and $\bar{R}_{QNoC}$, respectively). Other design decisions such as the qubit routing algorithm and resource allocation implementations complete the set of variables that will affect communication processes in our environment.

IV. On a Design Space Exploration for double stack communications-enabled Quantum Computers

DSE is a structured design methodology that allows to optimize a system maximizing a given cost function (or figure of merit) based on some parameters of interest [21], [22]. Like any other structured design process, this optimization relies on modeling the interdependencies among the different performance metrics and the variables describing the system. This modeling process might include analytic/theoretical expressions, behavioral models, computer-based simulations, or their zone-wise combinations.

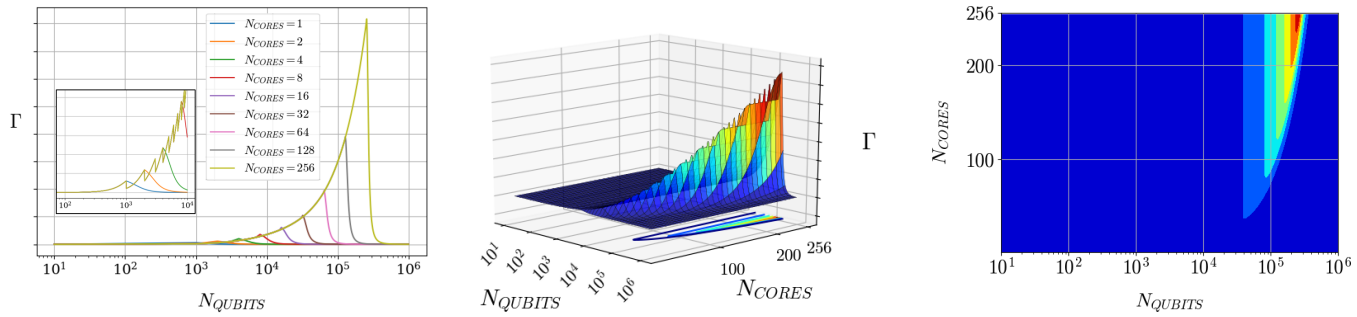
It is important to note that DSE is used to *design*, not just to *optimize* (performance metrics optimization is in fact just one of the DSE use cases – DSE is also useful for rapid prototyping or system integration with no need for analytical metrics [22]): whatever the design problem is, if the analysis is correctly prepared, the DSE analysis will not blindly look for "the extreme-case-highest-performing scenario". Rather, the main virtue of DSE is to be able to consider system-wide trade-offs and different metrics that may also affect the design optimality. For example, a DSE analysis of a network deployment will not optimize the average throughput of the entire network, but will take into account deployment costs and qualitative characteristics such as network reliability or flexibility. DSE achieves this by letting the designer to concurrently sweep all the open variables in the design space –instead of "manually" tweaking them in a one-by-one approach–, and to consolidate several performance/cost/qualitative metrics into a single merit figure Γ, which is then optimized.

The advantages of this methodology are thus threefold: *i)* Exploring the entire design space without being limited by the "intuition" and designer's previous experience that might hinder the way to the optimal (but maybe not intuitive) solution, *ii)* Providing not just a single optimal analytical solution, but rather design trends and guidelines extracted from the exploration, *iii)* Being valid also for early design decisions, when there are no analytical models or computer simulations for the performance metrics of the system.

Applying DSE to our specific problem implies identifying the variables that define a solution for our problem as well as the parameters that have to be accounted for when describing it in the multi-core quantum computing environment, and choosing one or more performance metrics (describing the computational power of the resulting design) to evaluate the whole multidimensional design space. We have already carried out a first description of the system parameters and open variables in our system design (layer by layer, see Section III), but choosing a complete set of metrics is not a trivial task, and even more in the case of a young research field such as Quantum Computing.

V. DSE results

In order to illustrate the possibilities of DSE, we have generated some synthetic data. Applying intuitive models to it, we have executed an analysis looking for a first answer to some of the most interesting questions raised before: How will the quantum computer scale in number of qubits? Will multi-core approach unlock the current monolithic quantum computers' scalability bottlenecks? Does the inter-core communications technology affect the performance of multi-core quantum



(a) (b) (c)

Fig. 4. **Scalability analysis** (a) Quantum computer's overall performance is plotted against the number of qubits used in the system, for several configurations in terms of number of cores used. Qubit operations' fidelity $F = 99.9\%$, $\epsilon_C = 5\%$, $\epsilon_I = 0.1\%$, $N_Q^{LIM} = 1000$ and $w_i = 1, \forall i$. (b) and (c) Performance analysis when varying both the number of qubits and the number of cores in the quantum computer. The isolines in the plot let us know different configurations that provide the same performance.

computers? Let us look into the procedure used to perform this first analysis before commenting on the results.

Although we aim at developing a complete Figure of Merit (FoM) with exhaustive models, for this introductory paper, we have used intuitive yet useful performance metrics and models, which are aggregated into the FoM Γ shown in Eq. (1), i.e. it is a preliminary example for illustrative purposes on the usage of DES for multi-core quantum computer design. For that reason, not all of the previously described architectural parameters have been included and the application of DSE is very straightforward, without leveraging all of its advantages. As a behavioral model, this first attempt suffices for showing all the possibilities that DSE has to offer.

$$\Gamma = \frac{w_{Qb} J_{Qb} \cdot w_{QF} J_{QF}}{w_F J_F \cdot w_I J_I \cdot w_C J_C} \tag{1}$$

where

$$J_{Qb} = 2^{\tilde{N}_Q} - 1$$
$$J_{QF} = QF = \frac{\tau_c}{L_G}$$
$$J_F = 2 - f^{N_Q}$$
$$J_I = 1 + \frac{\epsilon_I N_Q}{N_{CORES}} \cdot \left(\frac{\mathcal{H}\left(N_Q - N_Q^{MAX}\right) \cdot N_Q}{N_Q^{MAX}} \right)^3$$
$$J_C = 2 - (1 - \epsilon_C)^{N_{USED}}$$

and

$$\mathcal{H}(x) = \begin{cases} 0 & \text{if } x < 0 \\ 1 & \text{if } x \geq 0 \end{cases}$$
$$N_Q^{MAX} = N_Q^{LIM} \cdot N_{CORES}$$
$$N_{USED} = \left(N_Q / N_Q^{LIM}\right)$$
$$w_{Qb}, w_{QF}, w_F, w_I, w_C \in (0, 1]$$

J_i and w_i correspond to the metrics taken into account and their respective weights. N_Q is the total number of qubits in system, while $\tilde{N}_Q$ is a normalized version in order to have $J_{Qb} \in [0, 1]$. N_Q^{LIM} is the maximum number of qubits that may be integrated into a single core without incurring in severe crosstalk, while N_Q^{MAX} is the aggregated maximum of qubits that may be integrated into N_{CORES} (the number of cores in the processor). f is the 2-qubit fidelity, ϵ_I is the cumulative error per qubit when integrated in a given core (accounting for cross-talk and derivatives), and ϵ_C is the error rate increase due to communications overhead when adding a core to the system. N_{USED} is the number of cores that contain active qubits (i.e. qubits that are being used in the given configuration). Finally, $\mathcal{H}(x)$ is the Heaviside step function.

The definition of the Figure of Merit accounts for different errors and overheads that may synthesize the effect of: i) the computational power of an increasing number of qubits in the system (J_{Qb}, described as an exponential dependence with N_Q), ii) the quality factor QF (J_{QF}), iii) the degradation of the aggregated fidelity when integrating more qubits (J_F), iv) the cross-talk and other physical impairments derived from integrating many qubits in the same chip/core (J_I, which depends on the saturation point N_Q^{MAX} and the qubit-to-qubit disturbance ϵ_I) and v) the communications overhead when using more than one core (which depend upon the qubit mapping algorithm and the communication technology employed; for the sake of simplicity, in this first model we group these effects under a simplified error rate ϵ_C, accounting also for the fidelity degradation in core-to-core communications).

Using these assumptions and models, in Fig. 4 we show a scalability analysis: the Γ values for a wide range of quantum computer configurations. Using realistic values for ion trap technology from [1] and [6], the fidelity has been set to $F = 99.9\%$, the gate latency $L_G = 5.4 \cdot 10^{-7} s$, the coherence time $\tau_c = 2 \cdot 10^{-1} s$, $\epsilon_I = 0.0001$, $\epsilon_C = 0.05$, $N_Q^{LIM} = 1000$, and $w_i = 1, \forall i$. This analysis gives us information on how scalable the multi-core approach may be and some design guidelines on quantum computer configurations for an optimal performance.

In the leftmost plot, a single-core quantum computer is compared to several multi-core configurations, for a total

number of qubits N_Q in the system varying from 10 to 10^6 qubits. For each configuration, the performance (Γ) follows a peaky bell-shape trend, with a maximum close to $N_Q = N_Q^{MAX}$ (the optimal configuration for that number of cores). Trying to integrate more than N_Q^{LIM} qubits in a single core causes a steep degradation of performance. The single-core processor is clearly exceeded by multi-core approaches. This first model does not capture a realistic communications overhead when using many cores, thus the performance is monotonically increasing in N_{CORES}: of course, this might change when considering refined communication models and mapping strategies, that may work better with a low number of cores and suffer from worse communications overheads. In the zoom-in, the saw-like profile in the performance curve can be clearly seen: whenever the optimal qubit distribution requires another core to be used (if available in the configuration), the extra comms overhead causes a steep fall in performance. This implies that the configuration with more cores is not always the best performing one. The center and right-most plots contain a complete input variables sweeping, with N_Q varying from 10 to 10^6 qubits, and N_{CORES} from 1 to 256. Observe that the more cores are present in the system, the narrower is the performance curve, always growing to larger N_Q.

Using this simple model, we can clearly draw three main conclusions: *i)* the QNoC approach is promising as a scalability enabler, *ii)* for every multi-core quantum computer configuration there exists an optimal working range (Γ over a certain minimum threshold), and *iii)* the N_Q^{LIM} parameter clearly constrains the performance of the configuration and thus we should consider it as a fundamental design variable. With more accurate data and models, we postulate that this type of analysis will effectively accelerate and optimize the research on Quantum Computing.

VI. Conclusions and future work

In this paper we have presented a double joint full-stack layered architecture for quantum computers that introduces communications in a multi-core approach (Quantum Network on Chip) as a scalability enabler for performance-unlocked quantum computing. We have also introduced a system-wide optimization proposal (using Design Space Exploration) that might facilitate once-for-all design guidelines unifying the still separated design technologies into a consolidated solution with optimal technologies and parameters for every situation. This will definitively allow to happen all the unprecedented advances expected in application fields such as pharmacology, internet security and big-data analysis that we expect, as well as those that we cannot even imagine.

An initial DSE scalability analysis has been presented using intuitive basic models that help us to imagine the world of possibilities that DSE enables. With the present and future work of all the quantum community in the models we need to improve and perfect the DSE (e.g. fidelity or coherence time dependencies on gate and qubit technologies, models relating qubit communication error rates with inter-network topologies or number of qubits per core, etc...) we will be able to elaborate future analysis including quantum computers benchmarks comparison and qubit technology gap analysis and provide design guidelines, with a special emphasis on self-specification of QNoC.

References

[1] S. Resch and U. R. Karpuzcu, "Quantum computing: an overview across the system stack," *arXiv preprint arXiv:1905.07240*, 2019.

[2] M. Martonosi and M. Roetteler, "Next steps in quantum computing: Computer science's role," *arXiv preprint arXiv:1903.10541*, 2019.

[3] S. Wehner, D. Elkouss, and R. Hanson, "Quantum internet: A vision for the road ahead," *Science*, vol. 362, no. 6412, p. eaam9288, 2018.

[4] T. D. Ladd, F. Jelezko, R. Laflamme, Y. Nakamura, C. Monroe, and J. L. O'Brien, "Quantum computers," *Nature*, vol. 464, no. 7285, pp. 45–53, 2010.

[5] R. Van Meter and C. Horsman, "A blueprint for building a quantum computer," *Communications of the ACM*, vol. 56, no. 10, pp. 84–93, 2013.

[6] N. A. of Sciences, *Quantum computing: progress and prospects*. National Academies Press, 2019.

[7] J. Preskill, "Quantum computing in the nisq era and beyond," *Quantum*, vol. 2, p. 79, 2018.

[8] C. Monroe, R. Raussendorf, A. Ruthven, K. Brown, P. Maunz, L.-M. Duan, and J. Kim, "Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects," *Physical Review A*, vol. 89, no. 2, p. 022317, 2014.

[9] M. Caleffi, A. S. Cacciapuoti, and G. Bianchi, "Quantum internet: from communication to distributed computing!" in *Proceedings of the 5th ACM International Conference on Nanoscale Computing and Communication*, 2018, pp. 1–4.

[10] K. R. Brown, J. Kim, and C. Monroe, "Co-designing a scalable quantum computer with trapped atomic ions," *npj Quantum Information*, vol. 2, no. 1, pp. 1–10, 2016.

[11] L. Vandersypen, H. Bluhm, J. Clarke, A. Dzurak, R. Ishihara, A. Morello, D. Reilly, L. Schreiber, and M. Veldhorst, "Interfacing spin qubits in quantum dots and donors—hot, dense, and coherent," *npj Quantum Information*, vol. 3, no. 1, pp. 1–10, 2017.

[12] L. Jiang, J. M. Taylor, A. S. Sørensen, and M. D. Lukin, "Distributed quantum computation based on small quantum registers," *Physical Review A*, vol. 76, no. 6, p. 062323, 2007.

[13] S. Sargaran and N. Mohammadzadeh, "Saqip: A scalable architecture for quantum information processors," *ACM Transactions on Architecture and Code Optimization (TACO)*, vol. 16, no. 2, pp. 1–21, 2019.

[14] N. Isailovic, Y. Patel, M. Whitney, and J. Kubiatowicz, "Interconnection networks for scalable quantum computers," in *33rd International Symposium on Computer Architecture (ISCA'06)*. IEEE, 2006, pp. 366–377.

[15] X. Fu, M. A. Rol, C. C. Bultink, J. Van Someren, N. Khammassi, I. Ashraf, R. Vermeulen, J. De Sterke, W. Vlothuizen, R. Schouten *et al.*, "An experimental microarchitecture for a superconducting quantum processor," in *Proceedings of the 50th Annual IEEE/ACM International Symposium on Microarchitecture*, 2017, pp. 813–825.

[16] R. Van Meter and S. J. Devitt, "The path to scalable distributed quantum computing," *Computer*, vol. 49, no. 9, pp. 31–42, 2016.

[17] N. C. Jones, R. Van Meter, A. G. Fowler, P. L. McMahon, J. Kim, T. D. Ladd, and Y. Yamamoto, "Layered architecture for quantum computing," *Physical Review X*, vol. 2, no. 3, p. 031007, 2012.

[18] L. Benini and G. De Micheli, "Networks on chips: A new soc paradigm," *computer*, vol. 35, no. 1, pp. 70–78, 2002.

[19] A. Pirker and W. Dür, "A quantum network stack and protocols for reliable entanglement-based networks," *New Journal of Physics*, vol. 21, no. 3, p. 033003, 2019.

[20] A. Dahlberg, M. Skrzypczyk, T. Coopmans, L. Wubben, F. Rozpedek, M. Pompili, A. Stolk, P. Pawełczak, R. Knegjens, J. de Oliveira Filho *et al.*, "A link layer protocol for quantum networks," in *Proceedings of the ACM Special Interest Group on Data Communication*, 2019, pp. 159–173.

[21] E. Kang, E. Jackson, and W. Schulte, "An approach for effective design space exploration," in *Monterey Workshop*. Springer, 2010, pp. 33–54.

[22] M. Gries, "Methods for evaluating and covering the design space during early design development," *Integration*, vol. 38, no. 2, pp. 131–183, 2004.