Characterizing the spatio-temporal qubit traffic of a quantum intranet aiming at modular quantum computer architectures

Rodrigo, Santiago; Spanò, Domenico; Bandic, Medina; Abadal, Sergi; Van Someren, Hans; Ovide, Anabel; Feld, Sebastian; Almudéver, Carmen G.; Alarcón, Eduard

DOI
10.1145/3558583.3558846

Publication date
2022

Document Version
Final published version

Published in
Proceedings of the 9th ACM International Conference on Nanoscale Computing and Communication, NANOCOM 2022

Characterizing the Spatio-Temporal Qubit Traffic of a Quantum Intranet Aiming at Modular Quantum Computer Architectures

Santiago Rodrigo
srodrigo@ac.upc.edu
Universitat Politècnica de Catalunya
Barcelona, Spain

Domenico Spanò
spndnc98p02c710a@studenti.unirc.it
U. Mediterranea di Reggio Calabria
Reggio Calabria, Italy

Medina Bandic
m.bandic@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Sergi Abadal
abadal@ac.upc.edu
Universitat Politècnica de Catalunya
Barcelona, Spain

Hans van Someren
j.vansomeren-1@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Anabel Ovide
anabel.ovide@gmail.com
University of Tartu
Tartu, Estonia

Sebastian Feld
s.feld@tudelft.nl
Delft University of Technology
Delft, The Netherlands

Carmen G. Almudéver
cargara2@disca.upv.es
Universitat Politècnica de València
Valencia, Spain

Eduard Alarcón
eduard.alarcon@upc.edu
Universitat Politècnica de Catalunya
Barcelona, Spain

ABSTRACT

Quantum many-core processors are envisioned as the ultimate solution for the scalability of quantum computers. Based upon Noisy Intermediate-Scale Quantum (NISQ) chips interconnected in a sort of *quantum intranet*, they enable large algorithms to be executed on current and close future technology. In order to optimize such architectures, it is crucial to develop tools that allow specific design space explorations. To this aim, in this paper we present a technique to perform a spatio-temporal characterization of quantum circuits running in multi-chip quantum computers. Specifically, we focus on the analysis of the qubit traffic resulting from operations that involve qubits residing in different cores, and hence quantum communication across chips, while also giving importance to the amount of intra-core operations that occur in between those communications. Using specific multi-core performance metrics and a complete set of benchmarks, our analysis showcases the opportunities that the proposed approach may provide to guide the design of multi-core quantum computers and their interconnects.

CCS CONCEPTS

• **Computer systems organization → Quantum computing**; **Multicore architectures**; • **Networks → Network performance analysis**; *Network simulations*; *Network on chip*.

KEYWORDS

Quantum Computing, Multicore Quantum Architectures, Traffic Characterization, Network Performance Analysis, Network-on-Chip

ACM Reference Format:
Santiago Rodrigo, Domenico Spanò, Medina Bandic, Sergi Abadal, Hans van Someren, Anabel Ovide, Sebastian Feld, Carmen G. Almudéver, and Eduard Alarcón. 2022. Characterizing the Spatio-Temporal Qubit Traffic of a Quantum Intranet Aiming at Modular Quantum Computer Architectures. In *The Ninth Annual ACM International Conference on Nanoscale Computing and Communication (NANOCOM '22), October 5–7, 2022, Barcelona, Spain.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3558583.3558846

NANOCOM '22, October 5–7, 2022, Barcelona, Spain

ACM ISBN 978-1-4503-9867-1/22/10…$15.00
https://doi.org/10.1145/3558583.3558846

1 INTRODUCTION

Quantum Computing theoretical capabilities are beyond doubt, and almost fifty years of research have prepared the stage for what some call the Second Quantum Revolution [14]: an unprecedented *quantum* leap in key areas such as cryptography, biochemistry, big data analysis, or artificial intelligence[21]. However, although during the last decade the advances in quantum computer prototypes are impressive, they are still struggling with integrating tens to hundreds of qubits [4, 12, 22], far away from the amount needed for fully-fledged systems. Hard engineering challenges related with the unstable nature of quantum states (quantum decoherence, need for per-qubit control, and others) are still today hindering the scalability of this technology [3].

As an alternative way to ongoing research on better qubit isolation and control leading to the integration of more qubits in the same chip, in the last years several groups have proposed putting together currently available small-sized computing nodes and making them work coordinately. At first they were theoretical approaches aiming at widening the research frontier, presenting two different approaches: distributed quantum computing, related to the development of the Quantum Internet [9, 11, 29], and short-range multi-core quantum computers on-a-chip [8, 18, 23, 30]. While large-scale Quantum Internet seems very promising, important

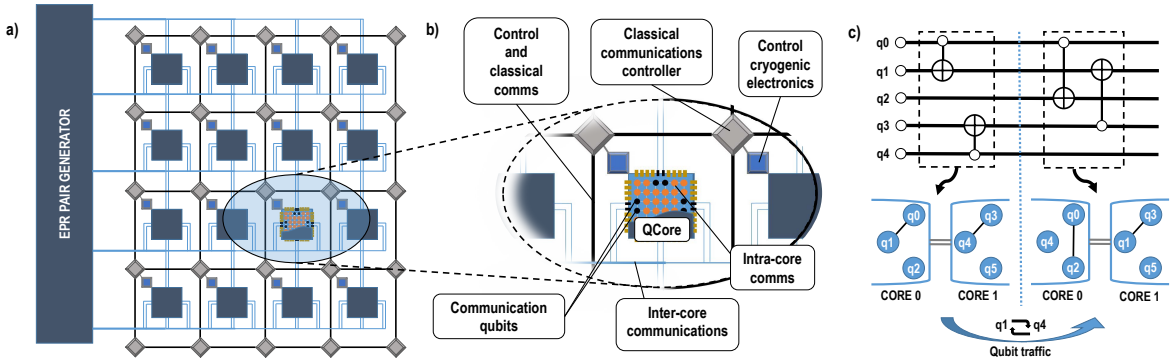


Figure 1: Multi-core quantum computing. a) 2D diagram of a multi-chip architecture. The classical network also depicted completes the networking infrastructure. b) Enumeration of the components, including intra- and inter-core communications. c) Qubit traffic generated by internal core topology and inter-core operations.

challenges (e.g. interfacing of qubits at cryogenic temperatures and communicating photons at room temperature) currently prevent its full development and the implementation of distributed quantum computing platforms. On the other hand, latest proposals on short-range communications linking existing NISQ processors include practical analysis and simulations on this type of architectures [26, 27]. Very relevantly, IBM has very recently released their updated quantum roadmap with multi-chip communication and teamwork computing as its keystone [16].

However, interconnecting quantum processors comes with its own share of hard challenges. Quantum data cannot be copied and latencies are crucial, as quantum decoherence steadily corrupts qubits. This, together with the need for computing with qubits in-place, calls for an entangled design between communications and computation in multi-core quantum computer architectures [25]. Such an approach should strive to minimize the communications overhead and optimize the computing efficiency when distributing qubits among different processors, since moving qubits around is expensive and affects directly the reliability of the whole computation. To that end, however, techniques for the analysis of the quantum communication traffic on these architectures is required.

In this paper, we aim to bridge this gap by providing a spatio-temporal analysis of qubit traffic, which allows to assess the impact of the quantum algorithm, mapping process and architecture on the actual execution. Based on this tool, we present some first results on traffic characterization of multi-core quantum architectures, which suggest that conscious design and optimization of compilers for multi-core quantum architectures could solve current issues in the amount and distribution (over time and space) of inter-core quantum data transfers.

In order to set a common ground for the discussion, we summarize in Section 2 the key concepts on quantum computing and communication. The use of traffic analysis on classical multi-core computing is reviewed in Section 3, discussing how this should be adapted into quantum computing and highlighting the need for a deeper understanding of multi-core quantum computers' inner workings in order to optimize these architectures. Section 4 is devoted to describing our qubit traffic analysis procedure, and in Section 5 we present some first results based on that tool, before concluding the paper.

2 BACKGROUND

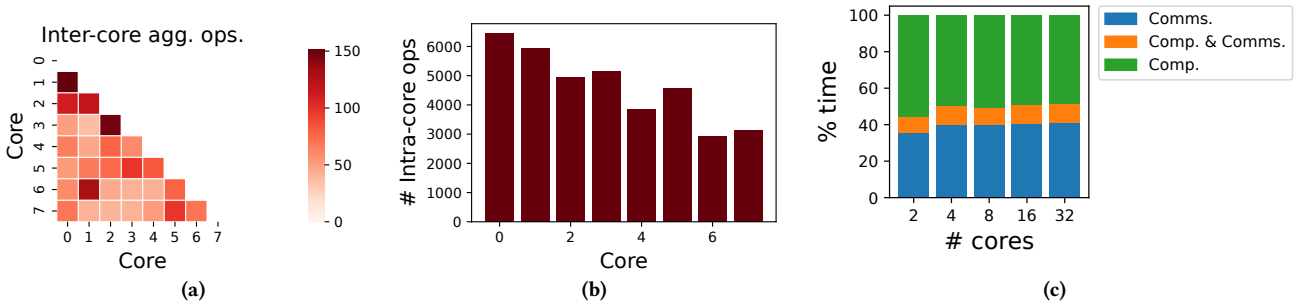
Reviewing the main concepts on multi-core quantum computing and communication lets us set the stage for our work and quickly highlight the most important "joints" where the design analysis of multi-core quantum computing should be stressed. For a deeper look into quantum computing and communications, the interested reader may refer to [9, 24].

2.1 On qubits, gates and circuits

The basic unit of quantum information is called a qubit. Similarly to its classical counterpart, a qubit can hold logical values, i.e. 0 and 1. However, due to *quantum superposition*, its quantum state can also be a linear combination of both 0 and 1 states. Moreover, when two qubits are superposed through specific quantum gates, the quantum state becomes a combination of 00, 01, 10 and 11 states, in a process that can be extended to an arbitrary number of qubits. In other words, a quantum computer with N superposed qubits is operating over 2^N states simultaneously, which provides an exponential increase in performance for certain applications.

When a qubit is measured, due to quantum mechanics' postulates, we obtain only a partial vision, and effectively *destroy* the quantum state: the qubit *collapses* into a deterministic value with certain probability. A derivative of this is the *no-cloning* theorem, which translates into the impossibility of duplicating a quantum state. Two or more qubits can also be *entangled*, i.e. whenever any of them is measured, all of them collapse into a definite state, with a non-zero correlation of the global result. For instance, two entangled qubits could be such that either both collapse to 0 or both collapse to 1. Along the present paper, we will differentiate *physical qubits* (the material holders of the quantum data, whether they are movable, e.g. photons, or not, e.g. spin qubits) and *virtual qubits* (the abstract quantum information which is operated on, swapped or teleported around the processor, measured, etc.).

These powerful properties allow for the *quantum* leap in computing performance of quantum processors, but also introduce high complexity in the design and operation of them. In particular, qubits should be as isolated as possible from the outer environment, as the quantum state they hold is unstable and tends to decohere after a certain time: being unfeasible to copy or regenerate them, a corrupt qubit implies an irreversible information loss.



(a) (b) (c)

Figure 2: Qubit traffic in multi-core quantum architectures a) Inter-core traffic, by pairs of communicating nodes. b) Intra-core operations (qubit gates) per core. c) Distribution of the execution time in computing and communicating in Cuccaro adder scaled in multi-core architectures. Observe the small amount of time in which computation and communications are allowed to be done in parallel.

Quantum programs (also called quantum circuits) are described using quantum gates, i.e. logical gates applied on either one or two qubits, which modify their quantum state. In order to improve their isolation and minimize decoherence, qubits are operated and measured in-place. However, when two qubits are required to interact by means of a two-qubit gate, their quantum state needs to be moved to adjacent positions in the computer. This involves a resource and time overhead, that may be critical for the overall execution performance.

2.2 A teamwork from quantum processors

Multi-core quantum architectures represent an effort of utilizing currently existing limited-size quantum processors into a large structure by interconnecting them and making the whole platform work jointly (see Fig. 1a). It may consist of dozens of quantum chips, containing hundreds to thousands qubits each, communicated both by a classical network (intended for signalling, message passing and measurement results communications) and a quantum backbone (to allow quantum state sharing).

Local operations on qubits are carried out as in single-core quantum processors. However, when a qubit from other node is required for an operation, a short-range quantum communication is performed. Although various types of techniques have been proposed, quantum teleportation is the most commonly assumed, due to its flexibility and robustness [10].

Teleporting a qubit between two cores is a process involving a pair of entangled qubits (specifically, an EPR pair [15]), shared among the two communicating nodes, and a classical channel among them. To perform the teleportation, some basic operations involving the qubit to be transmitted and the entangled qubit at the transmission side are applied, followed by a measurement. The result (a binary value) is then sent via the classical channel. With that information, the receiver can reconstruct the original transmitted quantum state by applying some corrections. Note that by being measured, the original state of a qubit is lost and hence the no-cloning theorem is respected. Observe also that the quantum information transfer is done without physically moving the qubit holding it, but rather through the distribution of the EPR pairs and the classical movement of measurement information.

Therefore, quantum teleportation needs at least three things to work: a classical network communicating all processors, an EPR pair generator and a quantum network connecting it with all the nodes (see Fig. 1b). Three different approaches may be used for the entanglement generation and distribution: *i)* sharing an EPR pair generator among all nodes, using Spontaneous Parametric Down-Conversion, and thus having a star-like topology connecting the nodes with it; *ii)* integrating on every core-to-core connection its own Bell State Measurement device, which is used to entangle photons coming from both transmitter and receiver nodes; *iii)* using entanglement at source, i.e. generating at the transmitter node the pair of entangled particles, and sending out a photon to associate the remote node to the entanglement [10]. For its low-resource requirements and simplicity, the first option is assumed for the rest of this paper.

3 USING TRAFFIC ANALYSIS FOR PERFORMANCE ANALYSIS

In classical multicore computers, the design of its internal Network-on-Chip (NoC) has become of extreme importance due to its impact on the performance of the entire processor. Since the design of any network requires an understanding of the traffic it needs to serve, considerable efforts have been spent over the years to characterize multicore systems and the applications that run on it.

Early works by Soteriou *et al.* [28] and Barrow *et al.* [6] analyzed a variety of multiprocessors between 16 and 32 cores running standard benchmark suites such as SPEC or PARSEC. In the former, the temporal burstiness, spatial hotspotness, and source-destination distance was studied, whereas in the latter, the focus was more on analyzing the memory sharing patterns leading to such traffic characteristics.

Subsequent studies pushed the analyses to larger systems up to 64 cores and delved into particular aspects, such as the time-varying characteristics of the traffic [7], which is often periodic as analyzed in [17]. This is due to the iterative nature of most algorithms running in multiprocessors, which further suggests that traffic is predictable. Further, the work in [1, 2] focused on multicast traffic only, demonstrating that such a subset of the workload is also bursty and predictable.

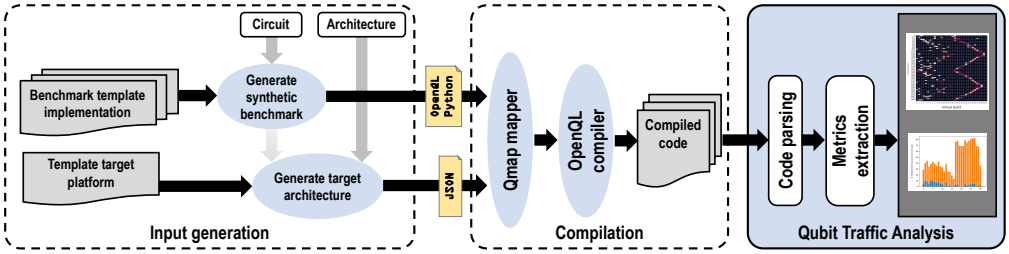


Figure 3: Flow diagram of the qubit traffic analysis tool

These workload characterization studies had several impacts on the NoC field. In particular, they allowed to *i)* study aspects such as the correlation between particular traffic characteristics and on-chip network congestion [17], *ii)* create synthetic traffic generators better reflecting real workloads for the evaluation of NoC designs [1, 7, 28], and eventually, *iii)* guide the design of improved topologies, routing policies, or congestion control mechanisms at the chip scale.

A pertinent question is then whether a similar approach can be used to characterize the workload of quantum processors. Before answering that question, though, it is important to see the main differences between both worlds. In classical systems, most of the traffic is an implicit consequence of the memory accesses produced by a multithreaded application and, hence, very hard to infer from compiled code. In quantum algorithms, on the other hand, communication primitives are explicit in the compiled code so that the traffic becomes not predictable, but rather known beforehand. Another difference is that due to the no-cloning theorem, it is hard to envisage the need for multicast communication at least resulting directly from the need to move the quantum state of qubits. Other than that, the metrics used in classical computing or the insight gained through analysis of its workloads, such as the iterative nature of communication, can be still useful in the quantum world.

4 A QUBIT TRAFFIC ANALYSIS SOFTWARE TOOL

Building a tool for analyzing traffic during the execution of quantum circuits in multi-core quantum architectures calls for firstly understanding where the traffic comes from and what are the main sources and stakeholders of traffic during the whole process.

In a generic multi-core quantum platform qubits are constantly moving around. Indeed, as stated in Section 2, whenever two distant qubits (even if they are on the same core) are to be operated by means of a two-qubit gate, they must be moved to adjacent positions. This implies that quantum circuits involve constant qubit traffic, both in and between cores. See Fig. 1c for a simple example on how qubit traffic is generated by both intra-core topology constraints and inter-core operations. In Figs. 2a and 2b, a simulated example based on a real quantum circuit is shown: specifically, the aggregated node-to-node and intra-node traffic of a sample execution of the QFT circuit of 128 qubits on a 8-core platform with 16 qubit per core. Observe the high total count of teleportations among cores, and the existence of some hotspots, attracting most of the communication and computation (cores 0 and 1).

In addition, inter-core communication is slower than intra-core communication operations: latencies are from 5× to 100× longer [5, 23]. This, together with the generally high dependency between gates, leads to almost idle execution intervals following high intensity ones. In the same example as before, see in Fig. 2c the time distribution of computation (execution of qubit gates) and communication (teleportation operations) when scaling Cuccaro adder circuit in multi-core architectures. Most of the time, the dependencies present in the circuit make the processor to idle while waiting for teleportations to end (in the example, only about 10% of the execution there are simultaneous computation and communication operations).

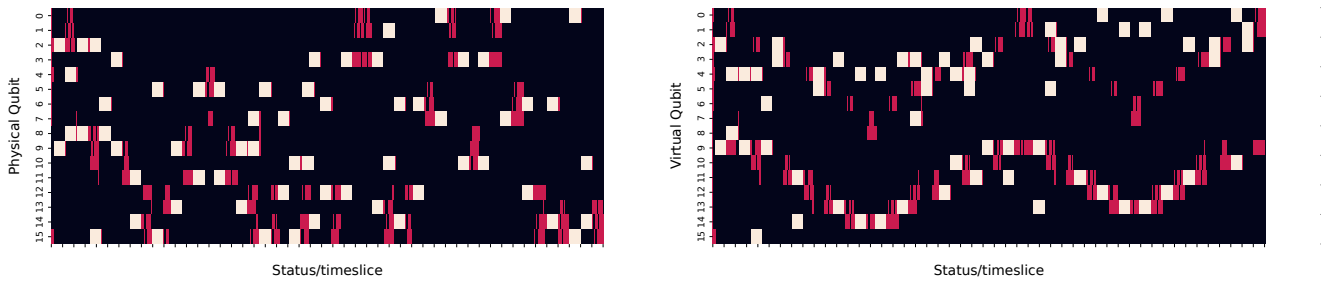
Therefore, knowing that all this communications overhead impacts on the reliability of the computation, we would desire to minimize these movements and equalize the traffic. Let us quickly review the three main stakeholders involved in traffic generation and control:

- **The quantum circuit**. The number and distribution of two-qubit gates will impact on the qubit traffic during execution.
- **The processor's topology**. A scarcely connected processor leads to a higher communications overhead when mapping two-qubit gates into the circuit. Moreover, in a multi-core scenario, the lower the ratio of number of qubits per core to number of cores, the higher the need for costly inter-core qubit communications.
- **The compiler algorithm (mapper and scheduler)**. When compiling the quantum circuit into a physical platform, optimizations can be applied to allow for minimizing the traffic overhead.

These would be the *deterministic* sources of traffic. There could be also some impact coming from communication errors during the execution, which could lead to unbounded communication latencies. However, we have decided not to include them into this work's analysis, so as to study them separately in future work including fully-fledged simulations.

Therefore, we have developed a software tool that, given a quantum circuit and a target many-core quantum platform, allows us to extract the qubit traffic by tracing all qubits along the execution and registering all the gates they participate in and the moves they are involved in.

The process, graphically explained in Fig. 3, consists on the following steps: *i)* generation of the quantum circuit with the corresponding qubit input length, *ii)* compilation of the quantum circuit on the target platform, always having the same number of physical qubits as the *width* (qubits involved) of the quantum circuit



(a) (b)

Figure 4: Execution trace of Grover's main routine for 16 qubits a) on physical qubits and b) on virtual qubits. Computation, communication, and idling times are represented in red, white, and black colors respectively.

and the required number of cores, and *iii)* parsing of the resulting cQASM code in order to obtain the trace of each of the qubits. That information may be used for analysis purposes studying traffic burstiness, hotspots and other related metrics.

For the compilation process, we have used OpenQL [19] and a modified version of the Qmap mapper [20] embedded in it, extended to the multi-core case following proposal from Baker *et al.* [5]. All the software has been programmed using Python 3.8.

4.1 Looking at a quantum circuit in a different way

Seeing the execution from this perspective gives us some insights on the performance of both the algorithm itself and the mapper, as we may easily observe how strong dependencies that block execution are dealt, the efficiency of the overall execution, idling periods, distribution of the job among the available cores and physical qubits, as well as analyzing the "life" of any virtual qubit along the execution.

Let us analyze a single example with our tool to see how these things may be observed and analyzed. We will use a small example for better visualization: the Grover's main routine for 16 qubits run in a 4-core platform (4 qubits per core). In Fig. 4 we present some graphical views of the execution to help us in the analysis. See in Fig. 4a the distribution of gate executions (in red) and teleportations (in white) versus the idling times (in black). The x axis is the timeline of the execution (the beginning at the top) and the y axis corresponds to the physical qubits in the system (ordered by core). The amount of dependencies among operations involves a high execution inefficiency, as most of the time qubits are idle, waiting for a single operation (either a quantum gate or a teleportation) to finish. The qubits in the bottom-most core are clearly accumulating the result and do the most computation and communication. In Fig. 4b, the y axis is now the virtual qubits, hence we see the logical operation of the algorithm more clearly. Now we can observe the dependencies better, and can differentiate virtual qubits that are expected to last with a coherent state for almost all the computation (e.g. qubit 9), whereas others are almost of no use (e.g. qubit 15).

These and other conclusions can be extracted using our tool, making it easier to give design guidelines for the algorithm, the architecture designer and the compiler engineer.

5 EXPERIMENTAL RESULTS

In the previous section we have focused on analyzing a single execution, seeing that several straightforward conclusions can be extracted. However, by aggregating these qualitative observations into numerical metrics and, in the future, using design space exploration techniques, we can obtain even more interesting insights.

5.1 Simulation set up and architectural space

In this paper, we introduce our tool and present some of the results obtained with it. Here we have restricted the exploration to a smaller set of examples, and will fully develop it on a larger upcoming paper. In particular, we have used multi-core architectures with the following fixed characteristics: the cores are interconnected via a classical network for exchanging classical messages and measurements. All nodes are connected via optical channels to an EPR pair generator instrumental for qubit teleportation. The teleportation operation is assumed by the compiler as deterministically time-bound (set to 1000 ns, around 4 times more than a SWAP gate, in order to stress the system [27]), and always performed as a SWAP operation, i.e. the two qubits involved are swapped after the teleportation operation. The connectivity inside every core is full, i.e. any qubit can perform a 2-qubit gate with any other in the same core. This is done to "isolate" in the analysis the inter-core communication from the intra-core computation, which does not involve extra-SWAPs. The rest of gate and qubit parameters are taken from typical superconducting flux qubits.

The exploration has been done by analyzing various algorithms (both real applications and random benchmarks, see below) and different platform configurations, varying the number of cores and number of qubits per core. In all cases, the circuits compiled in a given platform occupy all physical qubits available.

5.2 The selected algorithms

As benchmarks for assessment of communication overhead for multi-core architectures and their scalability, we opted for several algorithms that have potential to show computational advantage when run on quantum in comparison to classical computers, such as Quantum Fourier Transform (QFT), Grover's search algorithm and Cuccaro Adder. These algorithms, however, have a specifically defined structure which makes them scale with number of qubits in steady, sometimes even linear way (Grover's), in terms of their

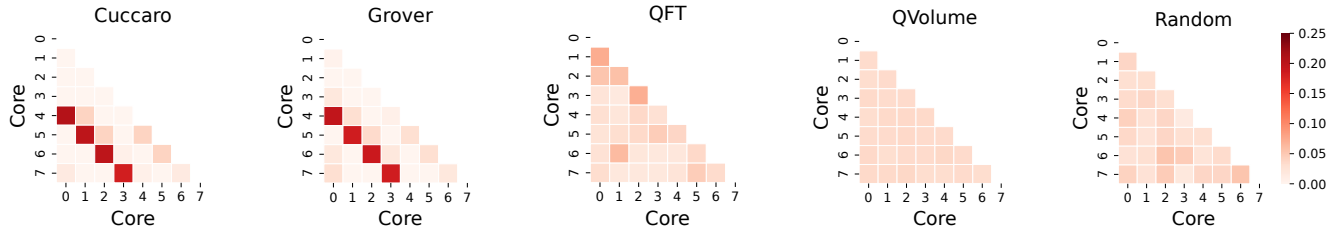


Figure 5: Inter-core traffic as the ratio of the number of teleportations for every pair of communicating nodes over total teleportations in all benchmarks assuming 8 cores with 16 qubits per core.

parameters like number of gates or two-qubit gate percentage. For that reason we additionally used randomly generated algorithms as well as quantum volume circuits [13], where we could have more influence on their parameters for any size of the circuit, and therefore probe our architecture in a worst-case scenario. The random algorithms we used were generated with uniformly chosen gates from a limited gate set with uniform distribution of those gates among qubits. Quantum volume circuits are used in general for probing even single-core architectures, as they are the most complex version of synthetic circuit with the highest two-qubit gate density (forces all qubits to be engaged in a two-qubit gate in each circuit layer).

5.3 Explorations

Following literature on traffic analysis for multi-core scenarios (see Section 3), we have performed our exploration on the selected benchmarks in a three-phase fashion: first, studying the temporal distribution of the quantum data trasnfers; then, focusing on their spatial distribution, and finally, summarizing both analysis in a spatio-temporal joint exploration.

For the temporal traffic distribution, we have studied the inter-core communication trends for the different algorithms. In Fig. 6, the moving average of the number of teleportations per timeslice in every circuit, together with the overall mean, is plotted for all algorithms. Two different cases are studied (8 and 16 cores, both with 8 qubits per core). Observe that both Cuccaro and Grover suffer from a high inter-core data transfer burst at the start, which may easily stress the system and cause a bottleneck on loaded or poorly-connected architectures. Both of them, together with QFT, have a quite low average number of teleportation (around 1), which is mostly related to the dependencies among operations in the code, forcing an almost linear, non-parallel, execution (as already observed in Fig. 2). Random and Quantum volume cases are good to stress the system, as they have more relaxed dependencies and allow for a higher teleportation rate. This communications requirements scale with the number of cores: this does not seem to be the case for Grover, Cuccaro and QFT, which may facilitate scaling on large multi-core architectures.

For the spatial traffic analysis, i.e. how evenly is the overall traffic distributed among the cores, we have focused on whether the compile circuit creates hotspots (cores attracting most communications). Hotspotness may be a natural consequence of most circuits, that e.g. concentrate the result on a given variable, but it results in network congestion. In Fig. 5, the inter-core traffic is presented for all benchmarks, for the 8 cores, 16 qubits per core case. The random and Quantum Volume cases are quite uniform, as expected, while that is also the case for QFT. Being a core part of some key quantum algorithms, avoiding network bottlenecks in QFT on multi-core architectures is relevant for their overall computational performance. Still, some minor hotspots (cores 0 and 1) can be detected, and probably further optimizations in the compiler could fix that. Grover and Cuccaro present a very similar behavior, which most probably has to do with the initial burst and flaws in the qubit mapping.

Finally, a joint spatio-temporal analysis is performed, using results as plotted in Fig. 7. A wider exploration is performed, for a core count ranging from 2 to 16 cores (8 qubits per core). We have used covariance (standard deviation σ over the mean) of both spatial and temporal traffic. For the spatial hotspotness we have used the number of teleportations per core over the whole execution, and for the temporal burstiness, we have used the number of teleportations per timeslice. Observe that there are two differentiate regions: random and Quantum Volume have low burstiness, while the rest are on the high burstiness end. Burstiness results in network inefficiencies due to unexpected bottlenecks and calls for overdimensioning the network capacity. See also that in general, the spatial hotspotness is specially high for Cuccaro, and that Grover scales quickly while QFT does it in a more controlled way.

6 CONCLUSIONS

In this paper we have substantiated the interest of qubit traffic analysis for efficient multi-core quantum architectures, and presented a tool for carrying out this characterization. We have showcased, with some first explorations, how traffic metrics may help in quantum algorithms classification, optimization of compilers for multi-core quantum architectures, and highlight the communications requirement for a given application and target architecture.

We plan to use this tool to do further explorations with larger sets of benchmarks and range of target architectures, as well as complementing this analysis with fully-fledged simulations that may shed the light on online quantum network management for error mitigation. Also, it is worth exploring which structural parameters in quantum circuits are the reason behind most inefficiencies found (data transfer bursts, hotspots, code dependencies...). This in-depth analysis might help us improve our inter- and intra-core communication strategy and later on give us the guidelines for multi-core device design that is more compatible with specific type of algorithms.

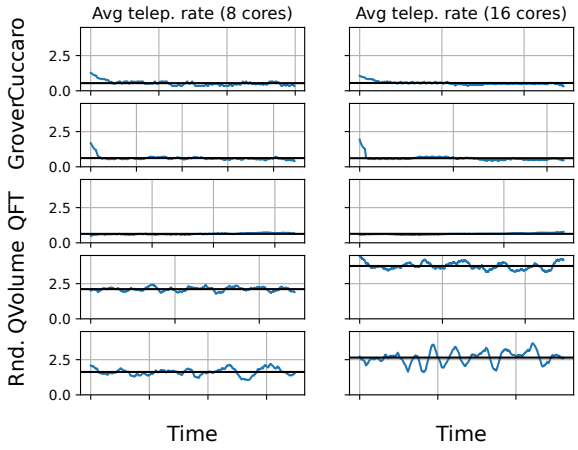


Figure 6: Average number of teleportations per timeslice in all benchmarks assuming 8 qubits per core and either 8 or 16 cores.

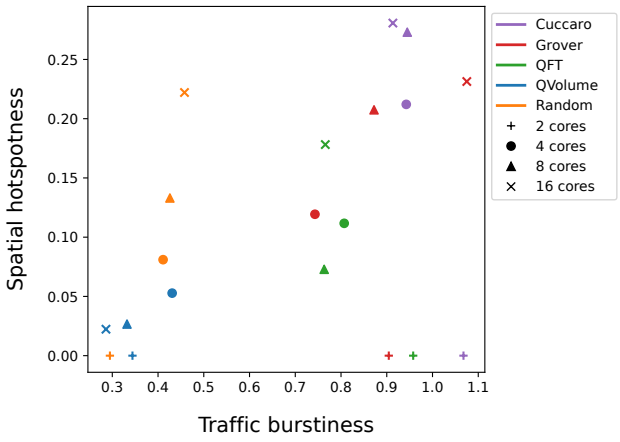


Figure 7: Summary of burstiness and hotspotness of all the evaluated benchmarks and core counts assuming 8 qubits per core.

REFERENCES

[1] Sergi Abadal, Raúl Martínez, Josep Solé-Pareta, Eduard Alarcón, and Albert Cabellos-Aparicio. 2016. Characterization and modeling of multicast communication in cache-coherent manycore processors. *Computers & Electrical Engineering* 51 (2016), 168–183.

[2] Sergi Abadal, Albert Mestres, Raul Martinez, Eduard Alarcon, Albert Cabellos-Aparicio, and Raul Martinez. 2015. Multicast on-chip traffic analysis targeting manycore noc design. In *2015 23rd Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. IEEE, 370–378.

[3] Carmen G Almudever, Lingling Lao, Xiang Fu, Nader Khammassi, Imran Ashraf, Dan Iorga, Savvas Varsamopoulos, Christopher Eichler, Andreas Wallraff, Lotte Geck, et al. 2017. The engineering challenges in quantum computing. In *Design, Automation & Test in Europe Conference & Exhibition (DATE), 2017*. IEEE, 836–845.

[4] F Arute et al. 2019. Quantum supremacy using a programmable superconducting processor. *Nature* 574, 7779 (2019), 505–510.

[5] Jonathan M Baker, Casey Duckering, Alexander Hoover, and Frederic T Chong. 2020. Time-sliced quantum circuit partitioning for modular architectures. In *Proceedings of the 17th ACM International Conference on Computing Frontiers*. 98–107.

[6] Nick Barrow-Williams, Christian Fensch, and Simon Moore. 2009. A communication characterisation of splash-2 and parsec. In *2009 IEEE international symposium on workload characterization (IISWC)*. IEEE, 86–97.

[7] Paul Bogdan and Radu Marculescu. 2011. Non-stationary traffic analysis and its implications on multicore platform design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30, 4 (2011), 508–519.

[8] Kenneth R Brown, Jungsang Kim, and Christopher Monroe. 2016. Co-designing a scalable quantum computer with trapped atomic ions. *npj Quantum Information* 2, 1 (2016), 1–10.

[9] A. S. Cacciapuoti, M. Caleffi, F. Tafuri, F. S. Cataliotti, S. Gherardini, and G. Bianchi. 2019. Quantum internet: Networking challenges in distributed quantum computing. *IEEE Network* 34, 1 (2019), 137–143.

[10] A. S. Cacciapuoti, M. Caleffi, R. Van Meter, and L. Hanzo. 2020. When entanglement meets classical communications: Quantum teleportation for the quantum internet. *IEEE Transactions on Communications* 68, 6 (2020), 3808–3833.

[11] M. Caleffi, A. S. Cacciapuoti, and G. Bianchi. 2018. Quantum Internet: From communication to distributed computing!. In *Proceedings of the 5th ACM International Conference on Nanoscale Computing and Communication*. 1–4.

[12] Jerry Chow, Oliver Dial, and Jay Gambetta. 2021. IBM Quantum breaks the 100-qubit processor barrier. *IBM Research Blog* (2021).

[13] Andrew W Cross, Lev S Bishop, Sarah Sheldon, Paul D Nation, and Jay M Gambetta. 2018. Validating quantum computers using randomized model circuits. *arXiv:1811.12926* (2018).

[14] Jonathan P Dowling and Gerard J Milburn. 2003. Quantum technology: the second quantum revolution. *Philosophical Transactions of the Royal Society of London. Series A: Mathematical, Physical and Engineering Sciences* 361, 1809 (2003), 1655–1674.

[15] Albert Einstein, Boris Podolsky, and Nathan Rosen. 1935. Can quantum-mechanical description of physical reality be considered complete? *Physical review* 47, 10 (1935), 777.

[16] Jay Gambetta. 2022. Expanding the IBM Quantum roadmap to anticipate the future of quantum-centric supercomputing. *IBM Research Blog* (2022).

[17] Paul Gratz and Stephen W Keckler. 2010. Realistic workload characterization and analysis for networks-on-chip design. In *The 4th workshop on chip multiprocessor memory systems and interconnects (CMP-MSI)*. Citeseer, 1–10.

[18] Nemanja Isailovic, Yatish Patel, Mark Whitney, and John Kubiatowicz. 2006. Interconnection networks for scalable quantum computers. In *33rd International Symposium on Computer Architecture (ISCA'06)*. IEEE, 366–377.

[19] Nader Khammassi, Imran Ashraf, J van Someren, Razvan Nane, AM Krol, M Adriaan Rol, L Lao, Koen Bertels, and Carmen G Almudever. 2020. OpenQL: A portable quantum programming framework for quantum accelerators. *arXiv preprint arXiv: 2005.13283* (2020).

[20] Lingling Lao, Hans van Someren, Imran Ashraf, and Carmen G. Almudever. 2019. Timing and resource-aware mapping of quantum circuits to superconducting processors. *arXiv preprint arXiv: 1908.04226* (2019). arXiv:1908.04226 [quant-ph]

[21] Margaret Martonosi and Martin Roetteler. 2019. Next Steps in Quantum Computing: Computer Science's Role. *arXiv preprint arXiv: 1903.10541* (2019).

[22] Doug McClure and Jay Gambetta. [n.d.]. Quantum computation center opens (IBM Research Blog). https://www.ibm.com/blogs/research/2019/09/quantum-computation-center/

[23] C Monroe, R Raussendorf, A Ruthven, KR Brown, P Maunz, L-M Duan, and J Kim. 2014. Large-scale modular quantum-computer architecture with atomic memory and photonic interconnects. *Physical Review A* 89, 2 (2014), 022317.

[24] National Academies of Sciences. 2019. *Quantum computing: progress and prospects*. National Academies Press.

[25] Santiago Rodrigo, Sergi Abadal, Eduard Alarcon, Medina Bandic, Hans Van Someren, and Carmen G Almudéver. 2021. On Double Full-Stack Communication-Enabled Architectures for Multicore Quantum Computers. *IEEE micro* 41, 5 (2021), 48–56.

[26] Santiago Rodrigo, Sergi Abadal, Carmen G Almudéver, and Eduard Alarcón. 2021. Modelling Short-range Quantum Teleportation for Scalable Multi-Core Quantum Computing Architectures. In *Proceedings of the Eight Annual ACM International Conference on Nanoscale Computing and Communication*. 1–7.

[27] Santiago Rodrigo, Medina Bandic, Sergi Abadal, Hans Van Someren, Eduard Alarcón, and Carmen G Almudéver. 2021. Scaling of multi-core quantum architectures: a communications-aware structured gap analysis. In *Proceedings of the 18th ACM International Conference on Computing Frontiers*. 144–151.

[28] Vassos Soteriou, Hangsheng Wang, and L Peh. 2006. A statistical traffic model for on-chip interconnection networks. In *14th IEEE International Symposium on Modeling, Analysis, and Simulation*. IEEE, 104–116.

[29] Rod Van Meter, Kae Nemoto, and W Munro. 2007. Communication links for distributed quantum computation. *IEEE Trans. Comput.* 56, 12 (2007), 1643–1653.

[30] LMK Vandersypen, H Bluhm, JS Clarke, AS Dzurak, R Ishihara, A Morello, DJ Reilly, LR Schreiber, and M Veldhorst. 2017. Interfacing spin qubits in quantum dots and donors — hot, dense, and coherent. *npj Quantum Information* 3, 1 (2017), 1–10.