TECHNISCHE UNIVERSITÄT DRESDEN
FAKULTÄT INFORMATIK

INSTITUT FÜR TECHNISCHE INFORMATIK
PROFESSUR FÜR RECHNERARCHITEKTUR

Routing on the Channel Dependency Graph: A New Approach to Deadlock-Free, Destination-Based, High-Performance Routing for Lossless Interconnection Networks

Kurzfassung

Dissertation
zur Erlangung des akademischen Grades

Doktor rerum naturalium
(Dr. rer. nat.)

vorgelegt von

Name:	Domke	Vorname:	Jens
geboren am:	12.09.1984	in:	Bad Muskau

Tag der Einreichung: 30.03.2017

Gutachter: Prof. Dr. rer. nat. Wolfgang E. Nagel, TU Dresden, Germany
Professor Tor Skeie, PhD, University of Oslo, Norway

1 Introduction

With Moore's law slowly coming to an end, many information technology domains started to scale-out. A global trend visible from small many-core systems-on-chip, such as the 256-core Kalray MPPA-256 chip [7] or the 1024-core Epiphany-V developed by Adapteva [20], over large-scale data centers or data wearhouses [19], to the world's most powerful supercomputers, such as the 40,960-node Sunway TaihuLight system hosting more than 10 million compute cores [9]. For high-performance computing (HPC) systems, it can be expected that the number of network endpoints will grow significantly [17] which emphasizes the role of the interconnection network as one of the most critical components in a supercomputer even more. These HPC networks, as well as other previously mentioned networks, are largely controlled by routing algorithms which determine how to forward packets. Routing algorithms have to balance multiple, partially conflicting, requirements. For example, they shall provide the best forwarding strategy to guarantee a certain quality of service (bandwidth, latency, etc.), which is an NP-hard problem in general, while minimizing the runtime of the routing in order to quickly react to failures of network components.

Unfortunately, as network topologies grow, failures of switches and connectors or cables become common. As opposed to network endpoints, the wiring complexity and infrastructural demands of cabling, e.g., arrangement of cable trays, make maintaining complex networks challenging and expensive. Thus, network structures are often kept in place over years and multiple generations of machines while other components such as CPU or main memory are upgraded. Today, many networks are based on the concept of over-provisioning the hardware, such as installing spare cables in the cable trays or having a spare parts inventory. Alternatively, they are operated in a deferred repair mode, in which a failed component will not be replaced instantaneously but within a reasonable time frame, such as a business day. Fail-in-place strategies are common in storage systems when maintenance costs exceed maintenance benefits, such as in large-scale data centers with millions of hard drives, e.g., IBM's Flipstone [1]. Hence, a fail-in-place approach for interconnection networks could be similarly beneficial, if supported by the correct combination of network topology and applied routing algorithm.

Routing algorithms for HPC systems have been the topic of many studies ranging from topology-specific routing algorithms [25, 35] through general deadlock-free algorithms [5, 28], more advanced deadlock-free algorithms balancing the routes [8], to advanced path-caching for quick failover [33]. A good overview is provided by Flich et al. [11]. Many advanced approaches for application-specific [16, 22] or topology-specific [14, 23, 31] routing and mapping assume idealized conditions such as a regular topology without faulty components, isolated bulk-synchronous applications communicating in synchronized phases, and the absence of system noise. Unfortunately, these assumptions are rarely true in practice. Presumably, integrating additional information about the current state of the supercomputer, such as state of the batch system (i.e., the current job mix) or application demands, can help guiding the network manager in assigning optimized routes on-demand which potentially increases utilization and achievable throughput.

Due to the previously addressed hardware failures and these on-demand software adjustments, topology-aware routing algorithms become increasingly inapplicable in modern supercomputers. Hence, avoiding deadlock situations [4, 6], a requirement for lossless Layer 2 networking, becomes much harder to achieve. Topology-aware routings usually avoid deadlocks algorithmically for many regular topologies, e.g., by restricting the routing to use only a subset of all available channel dependencies, as it is implemented by dimension-order routing (DOR) [24]. In contrast, many topology-agnostic routing technique for HPC and on-chip networks (NoC) use virtual channels to break deadlocks in arbitrary topologies [2, 8, 29, 30]. Yet, all these concepts have limitations: (1) topology-aware routings assume perfect topologies and often do not support switch/link failures, see Section 2, (2) cycle-avoiding routings often cannot balance routes and thus limit global bandwidth [26], and (3) routings based on virtual channel isolation fail when the required number of virtual channels is not available [11].

This thesis primarily focuses on flow-oblivious, static, destination-based, unicast routing for supercomputers, and is advancing the research field of HPC interconnects with the following contributions:

- Developing a network simulation and analysis framework which allows system administrators and designers to evaluate current, or plan future, fail-in-place networks and stipulate operation policies while taking failure rates into consideration;

- Showing, based on an exhaustive study of multiple InfiniBand (IB) routing algorithms and several real-world topologies, that fail-in-place network design can be accomplished with an appropriate combination of topology and routing algorithm while assuming a performance degradation up to a pre-defined threshold can be tolerated;

- Implementing a low-overhead scheduling-aware routing (SAR) for InfiniBand-based HPC systems, aimed at optimizing the effective edge forwarding indices and dark fiber percentage of realistic multi-user/multi-job HPC environments, and demonstrating its benefits over state-of-the-art routing mechanisms using simulations and communication benchmarks;

- Extending the formalism of per-packet consistent network updates for in-order delivery and deadlock-freedom, which is needed due to the frequent path reconfigurations performed by SAR, and show how to achieve both conditions using current InfiniBand hardware;

- Demonstrating the shortcomings of multi-step routing approaches with respect to creating deadlock-free forwarding rules and introducing a novel approach of routing on the complete channel dependency graph to overcome these issues; and

- Developing and implementing the deadlock-free, oblivious, and destination-based Nue routing (based on the novel approach) and showing its effectiveness, both with respect to the runtime calculating the forwarding rules and the achievable throughput resulting from the excellent path balancing, and proving its broad applicability to current and future interconnection technology.

2 Fail-in-Place High-Performance Networks

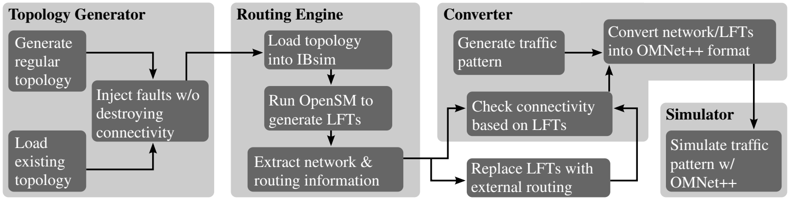
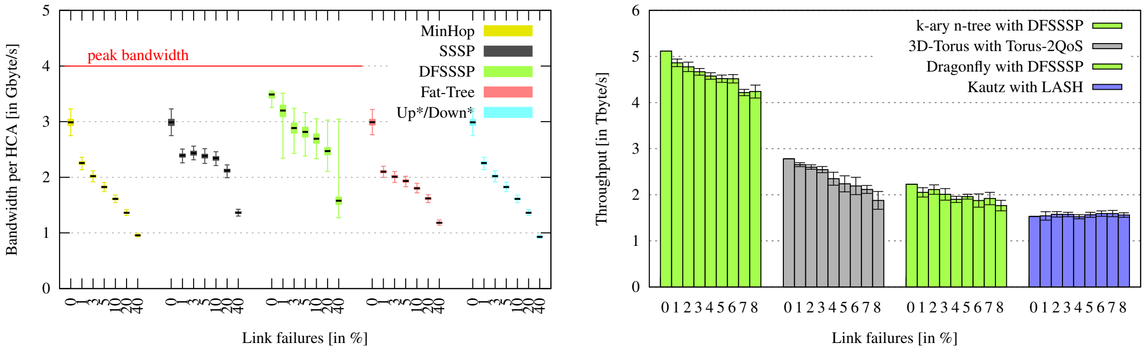

Topology Generator
Generate regular topology
Load existing topology
Inject faults w/o destroying connectivity
Routing Engine
Load topology into IBsim
Run OpenSM to generate LFTs
Extract network & routing information
Converter
Generate traffic pattern
Convert network/LFTs into OMNet++ format
Check connectivity based on LFTs
Replace LFTs with external routing
Simulator
Simulate traffic pattern w/ OMNet++

Figure 1: Toolchain to evaluate the network throughput of a fail-in-place network

As other high-performance computing hardware, networks are not immune to failures. According to previously published fault statistics, network failures constitute between 2% – 10% for the HPC systems at Los Alamos National Laboratory [27], over 20% for local area networks (LANs) of Windows NT based computers [15] and up to 40% for internet services [21] of the total number of failures. The failure data of the TSUBAME2 supercomputer [12] indicates an annual failure rate of ≈1% for InfiniBand links as well as 1% for the used switches, and shows the expected bathtub curve for hardware reliability [32]. Hence, assuming a constant annual failure rate of 1% is a fair representative failure rate to simplify further fail-in-place analysis and to extrapolate the achievable throughput degradation over a maximum system lifetime of eight years with the developed simulation framework shown in Figure 1. The fail-in-place property, in the context of interconnection networks, can be defined based on the differentiation between "critical" and "non-critical" network component failures. A critical component failure disconnects all paths between two hosts, whereas a non-critical failure only disconnects a subset of all paths between two hosts. The network fail-in-place strategy is to repair critical failures only, but continue operation by bypassing non-critical failures.

Conducted simulations show that two classes of deterministic routing algorithms are suitable to build fail-in-place interconnects: (1) fault-tolerant topology-aware routings, and (2) topology-agnostic routing algorithms. In terms of resiliency, topology-agnostic algorithms, such as (DF-)SSSP [8], are superior to topology-aware routing algorithms, such as fat-tree [36] or Up*/Down* [28], when the network



(a) Balanced 16-ary 2-tree with 256 HCAs

(b) Different networks with ≈2,200 HCAs

Figure 2: Left: whisker plots of consumption bandwidth for uniform random injection (three seeds); Right: histograms with error bars for all-to-all shift exchange pattern (ten seeds for fault injection)

suffers from a high percentage of failures, see Figure 2a. However, techniques to avoid deadlocks in topology-agnostic algorithms can limit their applicability for large networks, i.e., DFSSSP fails to route the Kautz(7,3) graph [18] with 2,352 attached terminals, see Figure 2b, when the number of virtual channels is limited to eight. This problem will be further addressed and solved in Section 4.

3 Utilization Improvement through Scheduling-Aware Routing

Idealized conditions, e.g., regular topology without faulty network components (Section 2), single application running on the whole system [14], or no system noise which is altering the message injection pattern, etc., usually do not apply to production supercomputers. An analysis of the job mix on two production supercomputers emphasizes the complexity of real-world installations. For example, approximately 66% and 50% of the compute jobs on Taurus [37] and TSUBAME2, respectively, are using multiple switches, while up to ≈100 jobs are executed simultaneously on these systems. Thus, the jobs potentially share the same network resources, meaning switches and inter-switch links, and therefore can influence each others communication performance. But at the same time, communications do generally not cross from one parallel applications to another, with the exception of management and I/O traffic targeted at special endpoints in the system, a fact ignored by state-of-the-art oblivious routings deployed in practice.

Performing a scheduling-aware routing (SAR) optimization for a supercomputer is only beneficial for parallel applications when their compute nodes are attached to multiple switches. Therefore, a filtering tool is needed, see Figure 3, which periodically polls SLURM [34] for the current state of the system, and filters out a list of multi-switch jobs which is then passed to the InfiniBand fabric manager. SAR routing is implemented as an extension for the topology-agnostic DFSSSP routing to include knowledge about batch job locations, which enables the algorithm to optimize the path calculation for intra-job paths. SAR's effectiveness is determined by two new metrics: (1) the effective edge forwarding index (extending the established EFI [3]) to differentiate between "useful" intra-job paths and unused inter-job paths, and (2) the proportion of superfluous links (or "dark fiber") to links usable for intra-job communication.

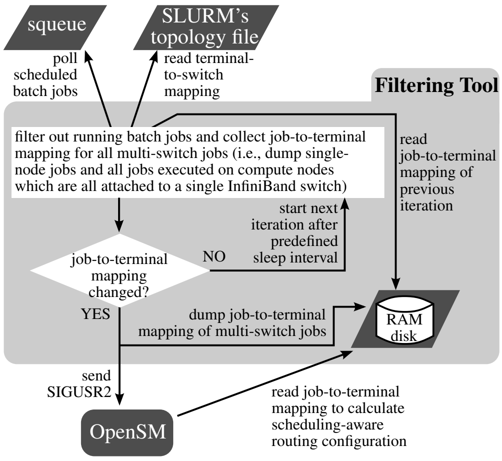

squeue
SLURM's topology file
poll scheduled batch jobs
read terminal-to-switch mapping
Filtering Tool
filter out running batch jobs and collect job-to-terminal mapping for all multi-switch jobs (i.e., dump single-node jobs and all jobs executed on compute nodes which are all attached to a single InfiniBand switch)
read job-to-terminal mapping of previous iteration
start next iteration after predefined sleep interval
job-to-terminal mapping changed?
NO
YES
dump job-to-terminal mapping of multi-switch jobs
RAM disk
send SIGUSR2
OpenSM
read job-to-terminal mapping to calculate scheduling-aware routing configuration

Figure 3: Flowchart of a filtering tool; Collect information about currently running multi-switch batch jobs and initiate recalculation of switches forwarding rules (linear forwarding tables) via SAR

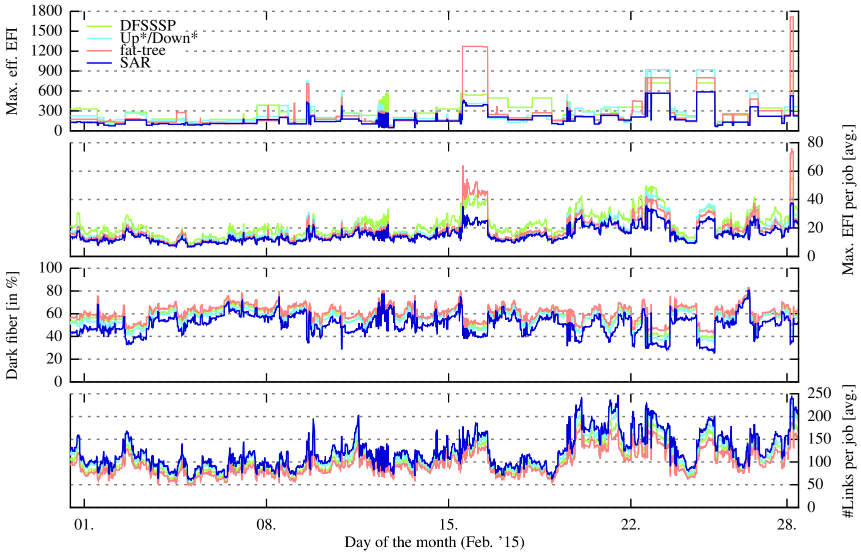


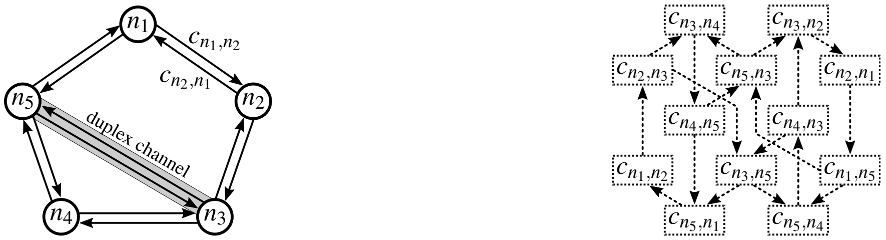
Figure 4: TSUBAME2 petascale supercomputer

Figure 5: Replay of job history TSUBAME2; Four routings applied per sampling point; Metrics collected: effective edge forwarding index (for individual jobs and sets of jobs) and dark fiber percentage

The results for the maximum effective EFI across all simultaneously executed jobs are shown in the top plot of Figure 5. As one can see, the scheduling-aware routing (blue line) outperforms the other three routing algorithms on the TSUBAME2 supercomputer. On average, SAR improves the effective EFI between 26.8% and 38.5% compared to the three competitors. The observed maximum difference is 71.2% between SAR and fat-tree routing. Similarly, the SAR approach lowers the maximal EFI per job (2nd plot), which should accelerate the achievable throughput within the jobs. Furthermore, SAR increases the number of links available to all running batch jobs by up to 17.74% for TSUBAME2 in comparison to the default fat-tree routing on this system, while also increasing the number of intra-job links. The scheduling-aware routing is now deployed for more than one year on the petascale Taurus production supercomputer at the TU Dresden to improve the system's utilization and fail-in-place properties.

4 Routing on the Channel Dependency Graph

A major problem for fail-in-place and other types of networks persists: No algorithmic approach is known to perform deadlock-free destination-based routing which is generally applicable to all technologies and topologies, and delivers better performance than Up*/Down*. Two general concepts exist for creating deadlock-free routing functions, both consisting out of two phases. The first concept is an analytical solution, also referred to as turn model, which basically restricts the usage of possible turns within the network as a first phase, either by knowledge of the underlying topology (e.g., dimension order routing [24]) or similarly to the Up*/Down* routing, before calculating the routes in the second phase. Unfortunately, this approach generally results in non-shortest paths and poor path balancing, and therefore



(a) 5-node ring network with shortcut

(b) Corresponding complete CDG $\overline{D}$

Figure 6: Complete CDG $\overline{D}$ for the 5-ring network with shortcut, see Figure 6a, assuming $k = 1$

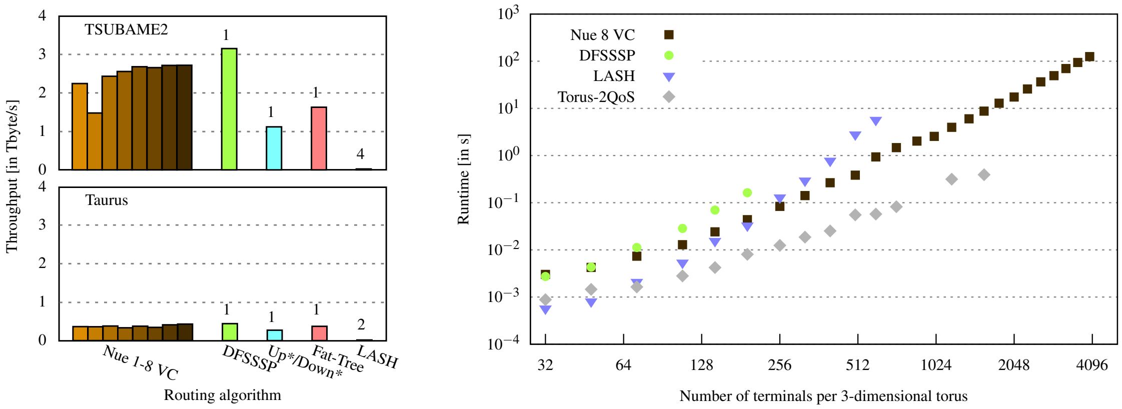
high latency and low throughput, respectively. In contrast, the current best practice of the second concept, e.g., as implemented by DFSSSP and LASH [30], is to decouple the two problems of (shortest-)path creation into phase 1 and deadlock-free assignment to virtual channels (VCs) into phase 2, because both problems require a different graph representation of the network and routes. Generally, this approach cannot be bound with respect to the number of virtual channels required to solve the deadlock problem, which is easily comprehensible by assuming a 5-node ring topology, e.g., similar to Figure 6a but without the shortcut between n_3 and n_5. Such a topology cannot be routed along shortest paths without at least two VCs. Hence, a two-phase approach is impractical for designing an universally applicable routing.

Assuming, it is possible to combine the information required to solve both problems within one graph, then this might allow to impose routing restrictions to the path creation on-demand, because the effects of a partial or full path on the channel dependency graph (CDG) can be checked simultaneously. Hence, it would be possible to avoid closing cycles in the CDG while calculating the paths instead of breaking the cycles later. This new graph, called complete CDG (see Figure 6b), includes all possible channel dependencies and represents one virtual layer [30]. A graph search algorithm, e.g., Dijkstra's algorithm, can traverse the graph and construct routes from all nodes to all other nodes and these routes are deadlock-free within this layer, see Algorithm 4.1. Furthermore, assuming the used network technology supports an arbitrary, but fixed, number of virtual channels, $k > 1$, then individual destination nodes can be assigned to different virtual layers. Hence, the graph search algorithm within one layer is able to calculate deadlock-free routes for all source nodes to the subset of destination nodes assigned to this virtual layer. Therefore, all routes in all virtual layers are deadlock-free without exceeding the virtual channel constraint.

This novel routing algorithm, called Nue routing, overcomes all limitations (1) – (3) mentioned in Section 1. To demonstrate the Nue's competitiveness, with respect to achievable global throughput, two real-world topologies, namely the multi-island fat-tree of Taurus and TSUBAME2's 2nd rail fat-tree network connecting 1,408 compute node, are analyzed, see Figure 7a. Nue routing is compared for a given topology to the other usable algorithms,

Algorithm 4.1: Nue routing calculates all paths within a network $I = G(N,C)$ for a given number of virtual channels $k \geq 1$

```
Input: $I = G(N,C), k \in \mathbb{N}$
Result: Path $P_{n_x,n_y}$ for all $n_x, n_y \in N$
Partition $N$ into $k$ disjoint subsets $N_1^d, \dots, N_k^d$ of destinations
foreach virtual layer $L_i$ with $i \in \{1,\dots,k\}$ do
    Create a convex subgraph $H_i$ for $N_i^d$
    Identify central $n_{r,i} \in N_i^H$ of $H_i$ via Brandes' algorithm
    Create a new complete channel dependency graph $\overline{D}_i$
    Define escape paths $D_i^\tau$ for spanning tree root $n_{r,i}$
    foreach node $n \in N_i^d$ do
        Identify deadlock-free paths $P_{\cdot,n}$ within $\overline{D}_i$
        Store these paths, e.g., in forwarding tables
        Update channel weights in $\overline{D}_i$ for these paths
```



(a) Simulated throughput for an all-to-all shift exchange pattern

(b) Routing runtime for 3D tori of various topology sizes with 1% link failures (missing dots = routing failed)

Figure 7: Comparison of Nue to other DL-free routings: throughput (7a) and runtime/applicability (7b)

for every number of virtual channels $1 \leq k \leq 8$. Besides the simulated throughput for each topology and routing combination, the number of needed virtual channels is given atop of the bars in Figure 7a. Two trends are visible for all investigated topologies: First, an increase of used virtual channels for Nue also increases the throughput for the all-to-all communication. The outliers from this pattern, e.g., the decrease in throughput for TSUBAME2 and two virtual channels, correlate to a sudden increase in fall backs to the escape paths. The second trend is that Nue shows a slight variance in throughput after reaching a certain peak, usually for about $k \geq 5$ in all investigated examples, but this generally depends on topology type and size. This behavior can be accounted to a mismatch between the static routing and the execution order of the point-to-point communications, which assemble the all-to-all traffic pattern, causing temporary congestion in the network which slows down the entire communication process [13]. In general, Figure 7a and further conducted simulations for other topologies show that Nue routing is competitive to the best performing routing for each individual topology, i.e., offers between 83.5% for a 10-ary 3-tree (with 1,100 terminals) and 121.4% throughput for a 1,536-node Cascade network [10] in comparison. Furthermore, two main conclusions can be drawn from the results shown in Figure 7b: Nue routing calculates the forwarding tables faster than the topology-agnostic DFSSSP, which has the same time complexity of $\mathcal{O}\left(|N|^2 \cdot \log |N|\right)$, and outperforms LASH routing for tori larger than 4x4x4 with 256 terminals attached. Only the topology-aware Torus-2QoS routing is on average 9x faster than Nue, which is as expected since Torus-2QoS is able to avoid deadlocks analytically, and therefore expensive cycle searches in the CDG are unnecessary. The second important result is an applicability of 100%, i.e., Nue routing scales with the topology size, while the other three deadlock-free algorithms fail.

5 Summary and Conclusion

The most important role of each supercomputer is to provide other research domains, such as material science, meteorology and climate research, biomedical informatics, or astrophysics, etc., with computational resources otherwise not available. The high-speed interconnection network, connecting the nodes of the HPC system, plays a crucial role in achieving scalability and throughput of these scientific applications. Key factors of the interconnection network are low latency, high bandwidth, high message throughput, deadlock-freedom, resiliency, and availability, hence factors which depend not only on the networking hardware, but also depend on the software used for network management. This thesis advances the state-of-the-art of network management in three main areas of interest for the HPC community, by: (1) showing that HPC interconnection networks can be operated in fail-in-place mode and providing a simulation toolchain to plan future systems and to establish operation policies, (2) developing a new and low overhead scheduling-aware routing algorithm to improve the scientific throughput and network utilization of multi-user/multi-job HPC environments, and (3) introducing a novel concept for solving the deadlock-freedom of routing algorithms, which allows the calculation of deadlock-free and path-balanced forwarding rules for arbitrary topologies within the given virtual channel constraint.

Even so the resulting irregular topologies of fail-in-place networks pose a challenge to the used routing algorithms, a low failure rate of the network components supports a fail-in-place network design strategy, which bypasses non-critical failures during the system's lifetime. However, both types of existing deterministic routing algorithms, i.e., fault-tolerant topology-aware and topology-agnostic, show limitations. The performance degradation using a topology-aware routing algorithm increases more with an increase of failures compared to topology-agnostic routings. In contrast, topology-agnostic routing algorithms are supposed to be ideal for fail-in-place networks, but the investigated algorithms either show weaknesses in terms of deadlocks, such as MinHop routing, or cannot be used beyond a certain network size. For example, the deadlock-avoidance via virtual channels by state-of-the-art routings can require more than the available number of virtual channels, e.g., in the case of LASH and DFSSSP for the 3-dimensional torus(7,7,7) with more than 2,000 terminal nodes, both exceed the limit of eight VCs.

While the fail-in-place network operation increases the resiliency and availability considerably, the overall achievable throughput decreases slightly when the supercomputer is used by a single scientific application. Nonetheless, this is rarely the case, since most HPC systems located at universities or national laboratories are used by multiple users simultaneously. The developed scheduling-aware routing (SAR) approach for the interconnection network is able to mitigate these problems. The conducted empirical study, based on the simulative replay of the historical job-to-node allocation of Taurus and TSUBAME2, reveals that SAR is able to improve the overall system utilization by up to 17.74%, and reduce the effective edge forwarding index of individual jobs by up to 71.2%. Conducted communication benchmarks on Taurus, while the system was simultaneously used by other researchers and scientific applications, showed throughput increases by up to 17.6% after SAR optimized the forwarding rules for the locality of the benchmark job.

Additionally, a fail-in-place network is also more prone to deadlocks, when it is based on lossless interconnection technology. However, the applicability of topology-aware and topology-agnostic routings for faulty regular topologies or arbitrary topologies is limited, as discussed before, and universally applicable routings, such as Up*/Down*, are scarce and perform poorly with respect to path length and balancing. A model implementation of the new approach of routing within the complete channel dependency graph, called Nue routing, is tailored for the deadlock-free, oblivious, and destination-based routing needed in Converged Enhanced Ethernet (CEE) and InfiniBand and can directly be employed for both, e.g., using InfiniBand's virtual lanes or using PFC together with Priority Code Point for CEE. Flit-level simulations indicate that Nue enables competitive or superior global throughput, i.e., between 83.5% and 121.4% throughput for an all-to-all traffic pattern, in comparison to the best performing state-of-the-art routing for the respective topology. Nue's characteristics and advantages, combined with its low time complexity of $\mathcal{O}(|N|^2 \cdot \log |N|)$ and memory complexity of $\mathcal{O}(|N|^2)$, make Nue routing a suitable algorithm to route modern large-scale supercomputers, lossless data center networks, and NoC architectures.

References

[1] M. Banikazemi et al. „Flipstone: Managing Storage with Fail-in-place and Deferred Maintenance Service Models“. In: *SIGOPS Oper. Syst. Rev.* 42.1 (Jan. 2008), pp. 54–62.

[2] T. Bjerregaard and S. Mahadevan. „A Survey of Research and Practices of Network-on-chip“. In: *ACM Comput. Surv.* 38.1 (June 2006).

[3] F. R. K. Chung et al. „The forwarding index of communication networks“. In: *IEEE Transactions on Information Theory* 33.2 (1987), pp. 224–232.

[4] E. G. Coffman Jr. et al. „System Deadlocks“. In: *ACM Computing Surveys* 3.2 (1971), pp. 67–78.

[5] W. J. Dally and C. L. Seitz. „Deadlock-Free Message Routing in Multiprocessor Interconnection Networks“. In: *IEEE Trans. Comput.* 36.5 (1987), pp. 547–553.

[6] W. Dally and B. Towles. *Principles and Practices of Interconnection Networks*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003.

[7] B. D. d. Dinechin et al. „A clustered manycore processor architecture for embedded and accelerated applications“. In: *2013 IEEE High Performance Extreme Computing Conference (HPEC)*. Sept. 2013, pp. 1–6.

[8] J. Domke et al. „Deadlock-Free Oblivious Routing for Arbitrary Topologies“. In: *Proceedings of the 25th IEEE International Parallel & Distributed Processing Symposium (IPDPS)*. Washington, DC, USA: IEEE Computer Society, May 2011, pp. 613–624.

[9] J. Dongarra. *Report on the Sunway TaihuLight System*. Tech Report UT-EECS-16-742. University of Tennessee, June 2016, pp. 1–24. (Visited on 12/01/2016).

[10] G. Faanes et al. „Cray Cascade: a Scalable HPC System based on a Dragonfly Network“. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. SC '12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, 103:1–103:9.

[11] J. Flich et al. „A Survey and Evaluation of Topology-Agnostic Deterministic Routing Algorithms“. In: *IEEE Transactions on Parallel and Distributed Systems* 23.3 (Mar. 2012), pp. 405–425.

[12] Global Scientific Information and Computing Center. *Failure History of TSUBAME2.0 and TSUBAME2.5*. Apr. 2014. (Visited on 04/01/2014).

[13] T. Hoefler et al. „Multistage Switches are not Crossbars: Effects of Static Routing in High-Performance Networks“. In: *Proceedings of the 2008 IEEE International Conference on Cluster Computing*. IEEE Computer Society, Oct. 2008.

[14] A. Jokanovic et al. „Quiet Neighborhoods: Key to Protect Job Performance Predictability“. In: *2015 IEEE International Parallel and Distributed Processing Symposium (IPDPS)*. Hyderabad: IEEE, May 2015, pp. 449–459.

[15] M. Kalyanakrishnam et al. „Failure Data Analysis of a LAN of Windows NT Based Computers“. In: *Proceedings of the 18th IEEE Symposium on Reliable Distributed Systems*. SRDS '99. Washington, DC, USA: IEEE Computer Society, 1999, pp. 178–.

[16] M. A. Kinsy et al. „Application-aware deadlock-free oblivious routing“. In: *Proceedings of the 36th annual International Symposium on Computer Architecture*. ISCA '09. New York, NY, USA: ACM, 2009, pp. 208–219.

[17] P. Kogge et al. *ExaScale Computing Study: Technology Challenges in Achieving Exascale Systems*. Tech. rep. TR-2008-13. Department of Computer Science and Engineering, Notre Dame, Indiana: University of Notre Dame, Sept. 2008.

[18] D. Li et al. „Graph-Theoretic Analysis of Kautz Topology and DHT Schemes“. In: *NPC*. 2004, pp. 308–315.

[19] Microsoft Corporation. *Whitepaper: Microsoft’s Cloud Networks*. Tech. rep. 2015. (Visited on 01/15/2017).

[20] A. Olofsson. *Whitepaper: Epiphany-V: A 1024 processor 64-bit RISC System-On-Chip*. Tech. rep. 2016. (Visited on 01/15/2017).

[21] D. Oppenheimer et al. „Why do internet services fail, and what can be done about it?“ In: *Proceedings of the 4th Conference on USENIX Symposium on Internet Technologies and Systems*. Vol. 4. USITS ’03. Berkeley, CA, USA: USENIX Association, 2003.

[22] M. Palesi et al. „A Methodology for Design of Application Specific Deadlock-free Routing Algorithms for NoC Systems“. In: *Proceedings of the 4th International Conference on Hardware/Software Codesign and System Synthesis*. CODES+ISSS ’06. New York, NY, USA: ACM, 2006, pp. 142–147.

[23] B. Prisacari et al. „Fast Pattern-Specific Routing for Fat Tree Networks“. In: *ACM Trans. Archit. Code Optim.* 10.4 (Dec. 2013), 36:1–36:25.

[24] T. Rauber and G. Rünger. *Parallel Programming - for Multicore and Cluster Systems*. Springer, 2010.

[25] G. Rodriguez et al. „Oblivious routing schemes in extended generalized Fat Tree networks“. In: *IEEE International Conference on Cluster Computing and Workshops, 2009 (CLUSTER ’09)*. Aug. 2009, pp. 1–8.

[26] J. C. Sancho et al. „A new methodology to compute deadlock-free routing tables for irregular networks“. In: *Network-Based Parallel Computing. Communication, Architecture, and Applications. 4th International Workshop, CANPC 2000. Proceedings (Lecture Notes in Computer Science Vol.1797)*. Berlin, Germany, 2000, pp. 45–60.

[27] B. Schroeder and G. Gibson. „A Large-Scale Study of Failures in High-Performance Computing Systems“. In: *IEEE Transactions on Dependable and Secure Computing* 7.4 (Dec. 2010), pp. 337–350.

[28] M. D. Schroeder et al. „Autonet: A High-speed, Self-Configuring Local Area Network Using Point-to-Point Links“. In: *IEEE Journal on Selected Areas in Communications* 9.8 (Oct. 1991).

[29] K. S. Shim et al. „Static virtual channel allocation in oblivious routing“. In: *Proceedings of the 2009 3rd ACM/IEEE International Symposium on Networks-on-Chip*. NOCS ’09. Washington, DC, USA: IEEE Computer Society, 2009, pp. 38–43.

[30] T. Skeie et al. „Layered Shortest Path (LASH) Routing in Irregular System Area Networks“. In: *IPDPS ’02: Proceedings of the 16th International Parallel and Distributed Processing Symposium*. Washington, DC, USA: IEEE Computer Society, 2002, p. 194.

[31] H. Subramoni et al. „Design of a Scalable InfiniBand Topology Service to Enable Network-Topology-Aware Placement of Processes“. In: *Proceedings of the International Conference on High Performance Computing, Networking, Storage and Analysis*. SC ’12. Los Alamitos, CA, USA: IEEE Computer Society Press, 2012, 70:1–70:12.

[32] A. Verma et al. *Reliability and Safety Engineering*. Springer Series in Reliability Engineering. Springer, 2010.

[33] J. C. Villanueva et al. *Whitepaper: Routing and Fault-Tolerance Capabilities of the Fabriscale FM compared to OpenSM*. Tech. rep. Fabriscale, July 2015, p. 10. (Visited on 03/16/2016).

[34] A. B. Yoo et al. „SLURM: Simple Linux Utility for Resource Management“. In: *Job Scheduling Strategies for Parallel Processing: 9th International Workshop, JSSPP 2003, Seattle, WA, USA, June 24, 2003. Revised Paper*. Ed. by D. Feitelson et al. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 44–60.

[35] E. Zahavi. „Fat-tree Routing and Node Ordering Providing Contention Free Traffic for MPI Global Collectives“. In: *J. Parallel Distrib. Comput.* 72.11 (Nov. 2012), pp. 1423–1432.

[36] E. Zahavi et al. „Optimized InfiniBand fat-tree routing for shift all-to-all communication patterns“. In: *Concurr. Comput. : Pract. Exper.* 22.2 (Feb. 2010), pp. 217–231.

[37] ZIH, TU Dresden. *Taurus - Network configuration*. Jan. 2013.