Designing an Exascale Interconnect using Multi-objective Optimization

Jose A. Pascual, Joshua Lant, Andrew Attwood, Caroline Concatto,
Javier Navaridas, Mikel Luján and John Goodacre
Advanced Processor Technology Group (APT)
The University of Manchester
Manchester, United Kingdom (UK)
Email: {jose.pascual, joshua.lant, andrew.attwood, carol.concatto,
javier.navaridas, mikel.lujan, john.goodacre}@manchester.ac.uk

***Abstract*—Exascale performance will be delivered by systems composed of millions of interconnected computing cores. The way these computing elements are connected with each other (network topology) has a strong impact on many performance characteristics. In this work we propose a multi-objective optimization-based framework to explore possible network topologies to be implemented in the EU-funded ExaNeSt project. The modular design of this system's interconnect provides great flexibility to design topologies optimized for specific performance targets such as communications locality, fault tolerance or energy-consumption. The generation procedure of the topologies is formulated as a three-objective optimization problem (minimizing some topological characteristics) where solutions are searched using evolutionary techniques. The analysis of the results, carried out using simulation, shows that the topologies meet the required performance objectives. In addition, a comparison with a well-known topology reveals that the generated solutions can provide better topological characteristics and also higher performance for parallel applications.**

I. Introduction

Exascale computing is the next challenge for the supercomputing community aiming to deliver exaflop-capable systems. To achieve this computing power, these systems must be composed of millions of interconnected nodes (cores) in order to execute massive parallel applications. The way these nodes are connected determines the topology of the network and has a great impact on the theoretical (and real) performance of the system. There are many possible topologies that can be used to build parallel systems, such as the classic torus and tree or more recent proposals such as Dragonfly [1]. The properties of these topologies are well-known and are used to build state-of-the-art supercomputers such as the Blue Gene family [2], the future Summit supercomputer [3] or many systems from Cray Inc. [4]. Recently, randomly constructed topologies have been proposed that seem to have better properties than the classic ones [5].

The European Exascale System Interconnect and Storage project (ExaNeSt) [6] is currently designing and building a prototype architecture capable of reaching Exascale computation. The aim of ExaNeSt is to develop a system that can be scaled up to the tens of millions of interconnected low-power-consumption ARM cores [7] to solve large-scale scientific and big data problems. In order to support a system of this size ExaNeSt is confronted with the huge challenge of designing an interconnect able to meet very strict performance, resilience, and cost constraints for a range of computational challenges.

The ExaNeSt Interconnect is a multi-tier interconnect which, for the purpose of this paper, can be divided into two distinct parts. The lower tiers, which are physically fixed by means of boards and backplanes, and the higher tiers which are fully reconfigurable and use FPGA-based routers allowing the computing and networking elements to be rearranged, forming different topologies. This flexibility allows us to build topologies that connect routers directly among themselves or indirect/hybrid topologies using some external elements (such as custom 3rd party FPGA-based interconnects or standard off-the-shelf commodity switches).

This flexibility offers the possibility of favouring topologies that prioritise one (or several) of the following objectives: reach specific performance levels (i.e. applications with known communication patterns), achieve fault tolerance requirements or maintain certain cost-efficiency levels. However, the large size of the system together with its reconfigurable capability, leads to an extremely large design space that is impossible to be processed manually. This means that we either restrict our system to standard topologies as the ones mentioned above or we automatise the exploration of the search space by considering specific design constraints.

This is precisely the focus of this work, i.e. the development of a framework to guide the selection of network topologies that favours some specific characteristics. More specifically, we use multi-objective optimization strategies that allow rapid converge to high quality solutions. In particular, we evaluate two well-know evolutionary strategies with problem-specific crossover and mutation operators. The use of multi-objective optimization enables us to search for designs with several characteristics of interest. In the case of ExaNeSt, the main design aims are to maximise performance and fault tolerance, whilst keeping the cost of connectivity as moderate as possible. With this in mind, we have developed objective functions that maximize the bisection bandwidth and the path-diversity while minimizing the number of physical connections.

The solutions generated by the optimization framework are studied in order to determine which optimization strategy

provides the best results. We compare not only the numerical values of the objective functions but also some properties of the topologies generated. However these properties are static and do not fully reflect the behaviour of the system during real operation. For this reason, we extend our analysis to measure several metrics when simulating a realistic workload using INRFlow, a flexible simulation tool able to deal with arbitrary networks and workloads. We compare the optimized topologies against an irregular network. Results show that the framework is able to design network topologies with the desired characteristics, and that applications running in those designs perform the best.

The remainder of this paper is organized as follows: Section II details specifics about the ExaNeSt architecture focusing on the interconnect. Section III describes characteristics that should be considered to design it. Then we formulate the problem of generating network topologies as a multi-objective optimization problem describing the evolutionary techniques used (Sections IV and V). We then assess the benefits of our approach using the experimental framework explained in Section VI. These results are analysed and discussed in Section VII. We conclude in Section VIII with some conclusions and future lines of work.

II. Background and Motivation

In this section we describe the architecture of the ExaNeSt project that aims to build an exascale system. One of the main characteristics of this system, in contrast with traditional supercomputers, is the placement of storage devices physically close to the computing elements. This new approach will avoid excessive latency and energy consumption because data will be frequently available in local devices. However, such a novel storage organization comes with the challenge of implementing a single, unified interconnect to handle both storage and application traffic whilst at the same time maintaining system power and cost constraints. To address this challenge, the Exanest interconnect will require a well-designed topology, able to minimize latency and number of hops while providing high bisection bandwidth.

Figure 1 depicts a general overview of the ExaNeSt project architecture. The computing power of ExaNeSt is provided by a combination of low-power ARM cores [7] and hardware accelerators (FPGAs [8]) referred to as a *Computing Element* (CE), this is the minimum building block used in ExaNeSt. The next level of ExaNeSt is the *Chassis*, this is composed of six interconnected CE's using a backplane that delivers high-bandwidth connectivity (each CE is provided with 64 links), whilst reducing the costs and power consumption of external cables and transceivers. In addition, each chassis contains two routers with a number of variable links (L) that are used to interconnect multiple chassis. As these routers are implemented on FPGAs, L can vary in order to deliver interconnects with different characteristics. Up to 16 chassis can fit inside a cabinet. The complete system can be composed of up to 256 cabinets.

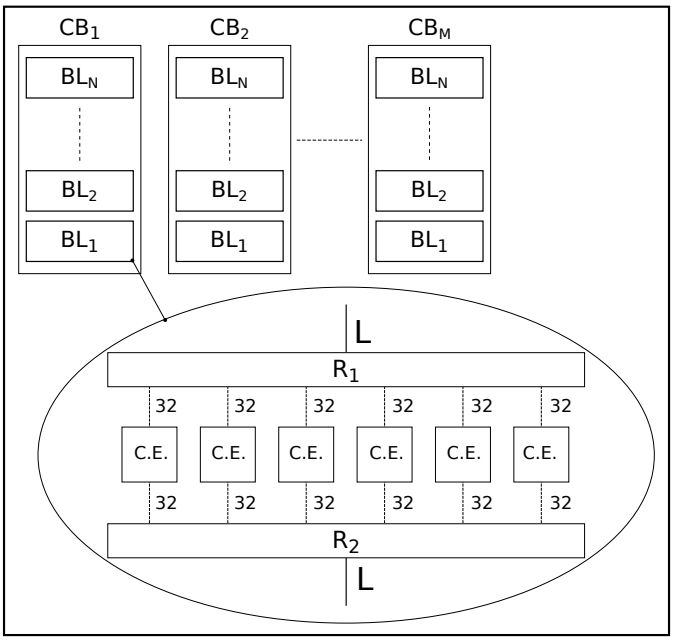


Fig. 1: Representation of the network architecture of the ExaNeSt project. It is composed of a number of cabinets (CB) that house several chassis (CH). Each of these chassis contains six computing elements and two routing elements (R) with up to L network interfaces (denoted as 'links' in our model below). Connections between routers are not restricted within the same cabinet being possible to directly connect routers in different CBs.

The interconnection among chassis can be performed following two alternatives: one is the use of a direct topology in which we connect the routers directly without using intermediate networking elements. In this scenario connections among chassis housed in different cabinets are allowed. The second alternative is to build an indirect topology in which the chassis are interconnected using either custom-made or off-the-shelf commodity switches. Looking at the interconnection architecture a third alternative, a hybrid network, could be easily constructed. In this case we would use a direct topology within the same cabinet and use Top-of-Rack (ToR) switches to interconnect multiple cabinets. In this paper, we restrict our framework to generate direct topologies for the sake of simplicity, but adapting it to generate other kind of topologies should be possible and is left as future work.

III. Modeling the Characteristics of an Interconnect

As explained in the previous section, ExaNeSt's flexibility allows for the construction of different arrangements among the chassis. In particular, we can select the number of links that are used to interconnect the networking elements. In this section we define characteristics desirable for interconnection networks which will be used to build a model. The aim is to use it to guide the design of the network topology. The

following three characteristics are key for the construction of the ExaNeSt interconnect:

Fault Tolerance: Fault tolerance is the property that enables a system to continue operating properly in the event one or more of its components fails. In the context of interconnects, we are mostly concerned with failures that prevent messages from reaching their destination due to single node failure.

Performance of the Applications: The final objective of ExaNeSt is the execution of large distributed (scientific and data-driven) applications. Therefore achieving high performance in the interconnect is of great importance. In order to reach this goal, low-diameter and high-bandwidth topologies together with high path-diversity between all sources and destinations are essential.

Deployment and maintenance cost: Another important characteristic is the cost of the interconnect. To keep this low, we should reduce the number of external cables and associated transceivers that are used to interconnect chassis. Moreover, reducing the number of components will also decrease the power consumption of the interconnect thus reducing operational expense.

These characteristics can be achieved by tuning the network parameters at design time and measuring the network's capability against the specific requirements listed above. However, since there is not just one metric that permits driving/modelling all given characteristics, we must define the minimal set that when put together cover each of these characteristics. These metrics are as follows:

The **Bisection Width** is the minimal number of links that need to be cut in order to split the system into two distinct parts. This metric is a good indicator of the performance (aggregate throughput) and the fault tolerance of the system (minimum number of link failures to physically disconnect the network).

The **Path-Diversity**, i.e., the number of non-overlapping, shortest paths among pairs of nodes, is a good indicator for both the expected performance of applications and the fault tolerance of the network. In terms of performance, the existence of multiple minimal paths between computing elements allows the avoidance of network congestion and the efficient use of network bandwidth. Regarding fault tolerance, the higher the number of paths to reach all the destinations the greater the ability of the system to continue working in case of multiple link failures.

The **Number of Links** used to interconnect the chassis. The relation of this metric with the cost and power consumption is clear. Reducing this metric could have a negative effect on the other two metrics. For the purpose of this paper, we define a link as a port in one router. Therefore a connection (cable) between two routers will be defined as a tuple with two links.

These three metrics are good indicators of the practical characteristics that we need for ExaNeSt and depend solely on the topology. For this reason we need to develop topologies in which these metrics are optimized. However, performing such optimizations by hand is challenging as the optimizations of these metrics conflict (see above). In the following two sections we describe the optimization techniques used to develop such topologies and how we implement the metrics to measure the properties.

IV. Multi-objective Optimization Algorithms

This section describes the optimization algorithms used in this work to solve the problem of the generation of network topologies, formulated as a multi-objective problem. We have selected two multi-objective evolutionary algorithms: NSGA-II and SMS-EMOA. We leave the evaluation of other strategies such as SPEA2 [9], HyPE [10] and NSGA-III [11] as part of the future work.

At each step of the optimization process (called generation), each of the algorithms maintains a set (population) of individuals (candidate solutions for a given problem). The quality of a solution is assessed using several fitness functions (objectives) that represent the (possibly constrained) problem; in this case, connections between different switches. Typically, at each generation the most promising individuals are chosen using a selection criterion and included in the new population. The rest of the population for the next generation (offspring) are then created by applying operators (e.g. crossover and mutation) to a set of individuals of the current population, to alter their structure and search the solution space. The optimization process is repeated until the stopping criterion is fulfilled.

The result of this optimization process is a set of solutions that simultaneously optimize each of the objectives (the Pareto set). The value of the functions achieved by the Pareto optimal solutions is called the Pareto front. Formally, we define a multi-objective optimization (minimization) problem subject to some restrictions as:

$$min\{f_1(x), \ldots, f_{N_{Obj}}(x)\} \tag{1}$$

$$\begin{cases} g_j(x) = 0 \ j = 1, \ldots, M, \\ h_j(x) \leq 0 \ j = 1, \ldots, K \end{cases} \tag{2}$$

where f_i is the $i-th$ objective function, x is a vector that represents a solution, N_{obj} is the number of objectives, $M+K$ is the number of constraints, and g_j and h_j are the constraints of the problem.

Now we explain each of the two optimization algorithms tested in this work. The main difference between them relies on the selection criterion used to choose the best candidate solutions at each generation.

Non-dominated Sorting Genetic Algorithm II: The aim of the NSGA-II [12] algorithm is to improve the adaptive fit of a population of candidate solutions to a Pareto front constrained by a set of objective functions. The population is sorted into a hierarchy of sub-populations based on the ordering of Pareto dominance. Similarity between members of each sub-group is evaluated on the Pareto front, and the resulting groups and similarity measures are used to promote a diverse front of non-dominated solutions.

S Metric Selection EMOA: SMS-EMOA selects the best candidate solutions using the *hypervolume indicator* (S metric) [13]. This measure is consistent with the concept of Pareto-dominance; the set of solutions with the highest value of the indicator dominates other sets. The algorithm's population evolves to a well-distributed set of solutions, thereby focusing on interesting regions of the Pareto front.

V. Defining the Optimization Problem

The aim of this work is to find suitable arrangements between the links provided by the network routers in order to define the topology among them. Taking as a starting point a random assignment among the links, we will use a multi-objective optimization algorithm to obtain designs with the desired properties. This optimization has the following objectives; to maximise the bisection width and increase the path-diversity whilst at the same time minimizing the number of links. In this section we focus on the formal definition of this particular optimization problem, as well as on the specific crossover and mutation operators needed by the two multi-objective optimization algorithms being evaluated.

A. Problem Definition

Given the description of the architecture of the chassis, i.e., the number of links per router R available to make the interconnections, the number of chassis per Cabinet $N \leq 16$ and the number of Cabinets $M \leq 256$, the link assignment involves finding a bijection φ between the set of links (L) that assigns each link $l \in L$ to other link $l' \in L$:

$$\begin{aligned} \varphi : & L \rightarrow L \\ & l \mapsto \varphi(l) = l' \end{aligned}$$

There are several possibilities to represent the solutions of this problem. The most simple would have the form $s = (s(1), \ldots, s(2 \times R \times N \times M)) = (l'_1, l'_2, \ldots, l'_{2 \times R \times N \times M})$ representing that the link i is connected with the link $s(i) = l'_i$. Solutions to this problem must obey two restrictions: connections among the same router and within the same chassis are not allowed because they are already connected internally.

Modelling the problem like this has three disadvantages: the size of the solutions space, the length of the solutions (up to 8192 $\times$ 2 $\times$ R for the largest possible system) and the fact that, if we do not put additional restrictions, we will obtain most likely completely irregular topologies. For these reasons we have developed an alternative way to model the problem in order to reduce those drawbacks.

The new model only considers the connections of one chassis and builds the rest of the network by symmetry. The new representation has the form $s = (s(1), \ldots, s(2 \times R)) = (l'_1, l'_2, \ldots, l'_{2 \times R})$ where:

- $s(i) \in [(R \times 2) - i + 1, (N \times R \times 2) - i]$ if the connection is made within the same cabinet (*internal link*) or
- $s(i) \in [(N \times R \times 2) - i + 1, (M \times N \times R \times 2) - i]$ for connections among different cabinets (*external links*).

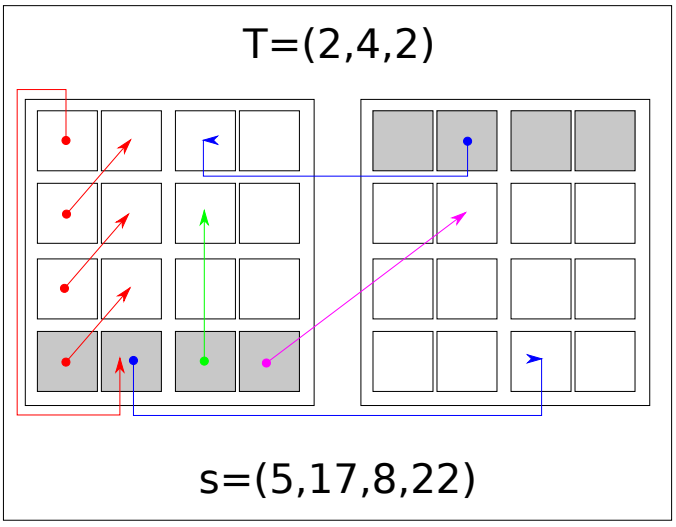


Fig. 2: Example of how an individual translates into a network topology. Red and green arrows represent internal links (same cabinet) while blue and pink arrows represent external links (different cabinets). For the sake of clarity we have only represented some of the connections.

This alternative representation reduces the size of the solution space, the length of the solutions and provides a degree of regularity to the network. In addition, it allow us to consider solutions with the form $s(i) = 0$ in order to leave links unused. Notice that due to the new representation, the parameter R translates into routers of size $2 \times 2 \times R$. It is possible that a solution generates an invalid topology because it is not fully connected. In that case it will be removed using a scheme detailed below.

In Figure 2 we have represented an example of how a solution translates into a real network topology. Given the problem parameters $T = (R = 2, N = 4, M = 2)$ and a solution $s = (5, 17, 8, 22)$, let us focus on the first chassis. Links 1 and 3 are connected to links (1+5) and (3+8) within the same cabinet because $s(1) = 5, s(3) = 8 < N \times R = 16$. The remaining red links indicate how the rest of the topology is populated ($5 \rightarrow 10$, $9 \rightarrow 14$, $13 \rightarrow 2$). The other links in the first chassis (green and pink) connect to the second cabinet ($s(2) = 17, s(4) = 22 \geq N \times R = 16$). We have also represented how the last chassis connects the second link (blue) with the first cabinet. For the sake of clarity we have represented just a few examples but the process to fully populate the network is the same.

As explained above, three major selection criteria will be considered to choose a link assignment. First, we favour solutions that maximize the bisection width and the path diversity. Both criteria try to positively impact the fault tolerance and the theoretical performance of the network. The third criterion reduces the cost because, otherwise, the solutions would use all the links in order to increase the other criteria. In addition, it indicates the amount of fault tolerance and connectivity that can be expected using different number of links. This is very

useful because, as we said in Section II, the use of FPGAs permits the implementation of a variable number of links in the deployment stage.

B. Objective Functions

More formally, we describe the link assignment as a three-objective optimization problem. Given B a function that returns the bisection width of a graph (defined in Section III), the first objective function to maximize is defined as follows:

$$f_1(s) : B(G) \tag{3}$$

where G is the graph that represents the topology. By maximizing B we increase the minimum number of links that must be broken to split the network in half. It is well-known that the bisection problem is NP-Hard. For that reason we have used the very efficient Kernighan/Lin heuristic algorithm. For further details check the original paper [14].

Given a solution s, we define the function $SDP(s, o, t)$ (Shortest Disjoint Paths) that returns the largest set of disjoint, minimal paths between an origin, o, and a target, t. Based on SDP, we define the second objective function to maximize as:

$$f_2(s) : 2 \times \frac{\sum_{i=1}^{|G|} \sum_{j=i+1}^{|G|} |SDP(a_i, a_j)|}{(|G| + 1) \times |G|)} \tag{4}$$

where $|G|$ is the number of routers and $a_i, a_j \in G$ and f_2 is the average number of paths between all pairs of routers in the topology. This function only considers disjoint paths.

The third objective to optimize is the number of links of the routers used to interconnect the chassis. Given a solution s we define the set of active links for this solution as $A^s = \{\exists i \in \{1, \ldots, 2 \times R\} \text{ s.t. } s(i) > 0\}$. Therefore the third function to minimize is defined as:

$$f_3(s) : |A^s| \tag{5}$$

C. Problem-specific Operators

As stated before, at each generation the optimization algorithms must evolve the current population using crossover and mutation operators. In this work, we have developed specific operators that consider the characteristics of the problem.

1) Guided Crossover Operator: Crossover is applied with probability P_{cross}. The crossover operator combines two individuals (or parents, pa_1 and pa_2) to generate two new candidates (or children, ch_1 and ch_2). First two points, $1 \leq c_1 \leq R$ and $R + 1 \leq c_2 \leq 2 \times R$, are generated uniformly at random. Then, elements 1 to c_1 and $R + 1$ to $R + c_2$ of pa_1 and pa_2 are exchanged to generate ch_1 and ch_2. At this point we need to check that the generated children are valid solutions by checking whether they represent valid topologies, i.e., are all the nodes connected. If not, we discard that solution (child).

2) Guided Mutation Operator: Mutation is applied with a probability P_{mut}. There are two types of mutation. The first type, selected with a probability of P_{new}, adds a new link into each router. Adding a link is carried out by selecting randomly one of the links equal to zero and replacing its value by a new one, v. With probability P_{ext}, we add an external link ($v \geq N \times R$), otherwise we add an internal link ($v < N \times R$). The second mutation, executed with probability $1 - P_{new}$, is to remove one link from each router. As with the crossover operator, removing links requires checking that the solution is still valid.

3) Solutions in the Pareto Front: The three-objective optimization algorithm generates a collection of solutions that represent multiple link assignments (Pareto set), with different trade-offs between bisection, path-diversity and number of links. As all Pareto optimal solutions are considered equally good, we can not decide which one is the best based on the outcome of our algorithm. However, after these three objectives are optimised, we can proceed to have a more detailed analysis of the generated topologies by looking at their performance in a more realistic scenario. In the next Section we will evaluate the solutions of the Pareto Set using our simulation environment, INRFlow.

VI. Experimental Framework

This section presents the simulation-based framework used to evaluate the topologies obtained from the optimization process. The experiments try to provide answers to two questions: (1) which optimization algorithm provides the best numerical results and (2) what kind of performance can we expect from the generated topologies.

A. Experiments to compare the optimization algorithms

The first step, to identify which of the two algorithms provides the best results, is to look directly at the values of the objective functions. To do that, we carried out two collections of 10 experiments that provide 10 different Pareto sets, from which we group the solutions with the same number of links.

In our experiments a cabinet is composed of 16 chassis and a router that can have up to 32 output ports ($R = 8$). We evaluate a small-system with a single cabinet–$T = (8, 16, 1)$–and allow internal links only ($P_{ext} = 0$). Then we assess a larger system with 4 cabinets–$T = (8, 16, 4)$. In order to observe the properties generated topologies exhibit when the proportion between adding new internal and external links changes, we consider the algorithms using three values of P_{ext}: 0.25, 0.50 and 0.75. The parameter configuration for the optimization algorithms is detailed in Table I. Note that for this particular work we have not made any effort to tune the parameters, and we have used the same values with all the optimization algorithms.

TABLE I: Parameter configuration for the optimization algorithms (NSGA-II and SMS-EMOA).

Parameter	Value	Description
N_{pop}	100	Number of individuals per generation
N_{gen}	100	Number of generations
P_{cross}	0.8	Probability of crossing operator
P_{mut}	0.8	Probability of mutation operator
P_{new}	0.5	Probability for mutation type
$P_{ext}{}^{\star}$	0, 0.25, 0.5, 0.75	Probability for external link

B. Experiments to evaluate the generated topologies

To corroborate the numerical results of the algorithms, we extend our evaluation to consider empirical results from our in-house developed simulator (INRFlow[1]). This tool models the behaviour of parallel systems, including the network topology, the applications and the workload generation. It takes as input the description of a topology (link arrangement) and measures several static properties (application-independent) and dynamic properties (with applications running). The static experiments are topology-specific, whereas the dynamic ones depend on the applications being run.

We have conducted a number of experiments using one application-like traffic model (we leave a more comprehensive study including other patterns and applications for future work due to paper length restrictions). The aim is to study the performance of the generated networks under a realistic scenario, in which real applications are executed. The workload selected provides characteristics from both high performance computing systems and data-centres. This captures the behaviour of a range of unstructured parallel applications, such as management traffic and graph-analytics applications that exhibit message causality and phases of higher and lower communication demands. For simplicity, the sources and destinations are chosen randomly and follow inter-message relationships.

These applications are executed in networks induced by the solutions of the optimization process. In particular, we have selected two solutions with 4, 8 and 12 links from $T = (8, 16, 1)$ and three solutions generated with $p_{ext} = (0.25, 0.50, 0.75)$ from $T = (8, 16, 4)$. The criterion to select the solutions has been the value of f_1 and f_2, selecting one with the highest bisection and another with the highest path-diversity. The networks are composed of 32 and 128 routers respectively and are evaluated in terms of execution time measured as the (simulated) time required to complete a workload (sending and receiving all messages).

We have compared the performance of the topologies against Jellyfish [5], a completely irregular network generated randomly. We denote it as (Irreg-r) where r is the size of the switches used. To ensure we exploit the maximization of the path diversity in all the topologies, we used Equal Cost Multiple Paths routing (ECMP) [15] which uses all the minimal paths among two nodes to send the traffic. For completeness, we compared it against Single Path routing (SP) which only employs one shortest path.

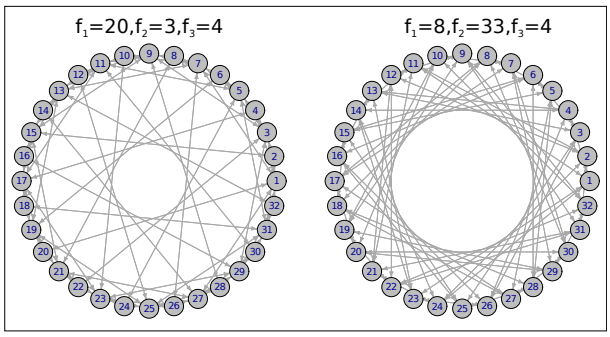
VII. Analysis of the results

In this section we report and analyse the results of the experiments explained in Sections VI-A and VI-B.

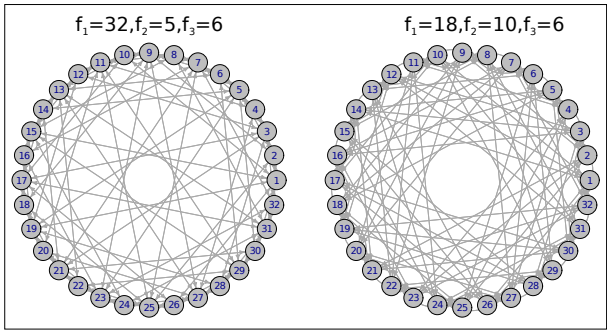
A. Evaluation of the Optimization Algorithms

The results from the generated topologies allow us to compare the performance of the optimization algorithms for this particular problem. Let us start analysing the results for the first collection of experiments for the problem instance $T = (8, 16, 1)$. Results are summarized in Table II, which gathers the mean, μ, and standard deviation, σ, of the multiple runs for both objective functions, f_1 (bisection width) and f_2 (path-diversity). Results seem to suggest that NSGA-II is better suited for the problem at hand. In a majority of cases, it is the one providing the highest bisection width for any number of links. Furthermore, in all cases it also obtains the highest values for f_2. The differences between the two algorithms are quite substantial in most of the cases. In Figure 3 we have depicted some examples of generated topologies together with their respective objective values.

[1]Available at: https://bitbucket.org/alejandroerickson/inrflow



(a) Routers of size 8



(b) Routers of size 12

Fig. 3: Graphical representation of four of the generated network topologies showing the values of the objective functions. For the sake of clarity we have represented routers with small size. Notice that solutions of size 4 and 6 translate into routers of size 8 and 12, respectively.

TABLE II: Means and standard deviations of objective functions f_1 and f_2 using NSGA-II and SMS-EMOA for the problem parameters $T = (8, 16, 1)$. The results are grouped by f_3 (number of links).

	NSGA-II				SMS-EMOA			
f_3	μ_{f_1}	μ_{f_2}	σ_{f_1}	σ_{f_2}	μ_{f_1}	μ_{f_2}	σ_{f_1}	σ_{f_2}
2	**6.04**	**1.06**	0.29	0.00	6.00	1.06	0.00	0.00
4	**15.96**	**10.16**	4.48	11.46	14.40	6.56	5.46	6.91
6	27.17	**8.57**	5.67	1.69	**30.38**	6.01	4.46	2.65
8	43.71	**11.99**	7.05	1.62	**44.50**	6.89	8.37	3.80
10	**59.54**	**10.91**	6.39	5.93	56.70	5.57	9.04	3.40
12	**73.88**	**7.00**	5.24	4.09	70.50	5.11	9.51	1.60
14	**87.88**	**7.53**	7.95	4.87	83.71	4.46	6.84	0.69
16	**101.07**	**6.89**	10.02	4.54	94.22	4.60	10.62	0.22

Now, let us analyse the results for the second collection of experiments for the problem instance $T = (8, 16, 4)$. Results are summarized in Table III for three values of P_{ext}. Again, we can see that NSGA-II achieves the highest values for both functions f_1 and f_2 for any value of f_3 in almost all of the cases. Moreover, the change in the probability p_{ext} barely affects the algorithms behaviour; NSGA-II still outperforms SMS-EMOA in most cases. We also wanted to assess how p_{ext} affects the objective functions and hence, the properties of the topologies. The results are not conclusive for this particular characteristics (they are similar for any value of P_{ext}) but could be affecting others such as the diameter or the average distance of the network.

In the last two rows we have summarized the average diameter and distance of all the networks generated by the Pareto set. Although these properties are not part of the optimization, they can be used as indicators of the quality of the topologies being generated. The solutions generated with $P_{ext} = 0.75$ and NSGA-II achieve the lowest diameter and average distance: 4.53 and 2.80. If we look at the values achieved by SMS-EMOA we can see that the lowest values of diameter and distance are achieved using $P_{ext} = 0.25$. For these particular network properties, both algorithms seem to behave similarly.

Note that the objective criteria used with our optimization algorithms only provide *hints* about the expected benefits for applications. The achievable benefits, expressed in more tangible terms, are analysed in the following section.

B. Evaluation of the Network Topologies

The objective functions f_1 and f_2 were designed to have a positive impact on both the performance of the applications and the fault tolerance, but we need to assess those impacts in a meaningful, measurable way. The impact of f_1 is clear, the higher the better, as this characteristic does not depend on applications. The impact of both f_1 and f_2 on the performance is not so clear and we want to evaluate it comparing the execution time of the selected application in several topologies.

In Figure 4 we have depicted the results obtained for some networks obtained from the Pareto set of the problem $T = (8, 16, 1)$ represented as (f_1-f_2-f_3). (22-2-4) and (8-3-4) are built using routers of size 8, (50-9-8) and (38,14,8) with routers of size 16 and (74-6-12) and (72-16-12) with routers of size 24. We compare them with the corresponding irregular network: Irreg-8, Irreg-16 and Irreg-24 respectively. The results clearly show that in all cases, applications executed in the topologies generated using optimization achieve the lowest execution time, even when using the SP routing.

Now we focus on the improvement achieved using ECMP. This was expected because one of the criteria to optimise was the path-diversity and ECMP makes use of them. results show that ECMP is rather beneficial. In all networks the use of multi-path routing improves the execution time of the applications, being especially remarkable the improvement for (22-2-4) and (8-3-4) in which execution times are more than halved.

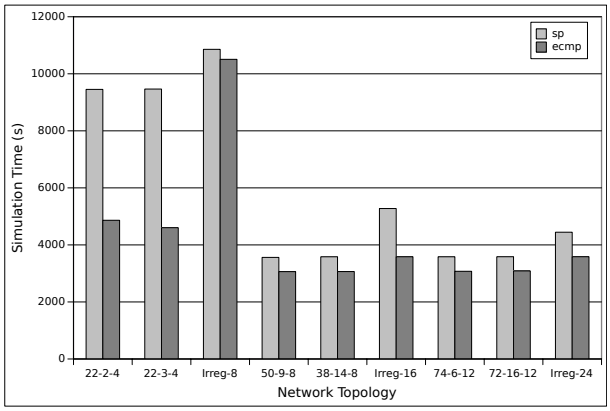


Fig. 4: Execution time of an unstructured application in multiple topologies using SP and ECMP routings. The network is composed of 32 routers (1 cabinet) with sizes 4, 8 and 12.

Let us focus now on the results obtained with $T = (8, 16, 4)$. In this case we have selected three solutions with routers of size 16: (106-13-8) from a Pareto set generated using $P_{ext} = 0.25$ and (124-12-8) and (150-11-8) from Pareto sets using $P_{ext} = 0.50$ and $P_{ext} = 0.75$ respectively. We have selected these particular solutions because they have a similar value of f_2. Our intention is to test whether there is a difference in performance between them. The results are depicted in Figure 5.

Looking at the results for the SP routing, the four networks require almost the same time to complete the execution. However, when we use the ECMP routing, the picture changes. The use of multiple paths doubles the performance of applications executed in our optimised topologies, but has less noticeable effect on the irregular network. Surprisingly, the execution time of our three networks is very similar, suggesting that, as intended, both f_1 and f_2 have a positive impact on the performance.

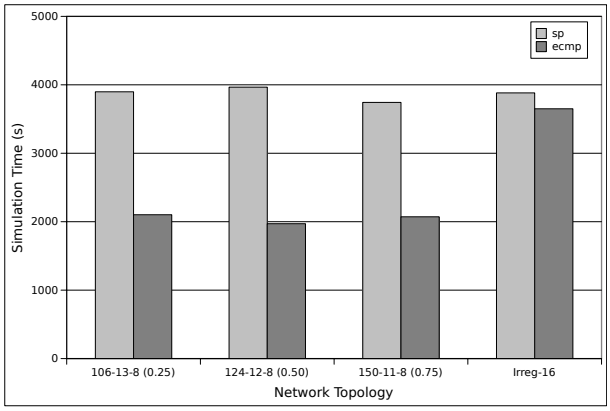


Fig. 5: Execution time of an unstructured application in four topologies using SP and ECMP routing policies. The network is composed of 128 routers (4 cabinets) of size 16.

TABLE III: Means and objective functions f_1 and f_2 using NSGA-II and SMS-EMOA for the problem parameters $T = (8, 16, 4)$ and three values for p_{ext}. The results are grouped by f_3 (number of links). The last two rows corresponds to the average diameter and distance of the whole Pareto set. For the sake of clarity we do not report the standard deviation.

	$p_{ext} = 0.75$				$p_{ext} = 0.50$				$p_{ext} = 0.25$			
	NSGA-II		SMS-EMOA		NSGA-II		SMS-EMOA		NSGA-II		SMS-EMOA	
f_3	μ_{f_1}	μ_{f_2}	μ_{f_1}	μ_{f_2}	μ_{f_1}	μ_{f_2}	μ_{f_1}	μ_{f_2}	μ_{f_1}	μ_{f_2}	μ_{f_1}	μ_{f_2}
2	**14.67**	**1.02**	12.67	**1.02**	13.00	**1.02**	**14.00**	**1.02**	**15.50**	**1.02**	13.33	**1.02**
4	40.00	**15.97**	**41.33**	6.21	**43.29**	**11.01**	37.66	9.07	42.00	**11.25**	**43.22**	5.41
6	**86.00**	**10.89**	81.83	7.42	76.24	**28.62**	**85.29**	6.33	**80.10**	**10.38**	78.63	7.68
8	**133.33**	**9.09**	122.11	7.28	121.57	**9.84**	**122.60**	8.38	**140.27**	**8.33**	132.00	5.90
10	157.67	**11.25**	**159.67**	7.36	**177.67**	**8.73**	165.30	6.74	**196.07**	**8.74**	163.11	5.62
12	**231.85**	**9.45**	214.44	6.13	**234.71**	**8.07**	228.33	5.46	**229.86**	**12.36**	196.00	6.89
14	279.82	**11.07**	**282.00**	6.01	**279.40**	**9.80**	274.40	6.06	**273.82**	**9.35**	236.13	6.95
16	**327.00**	**7.05**	299.50	5.97	**305.56**	**7.49**	302.50	5.95	**309.83**	**7.58**	307.00	5.79
Avg. Diam	4.53		5.00		5.10		4.72		4.71		4.58	
Avg. Dist	2.80		3.04		3.07		2.91		2.88		2.81	

VIII. Conclusions and Future Work

In this work we have presented an optimization framework to provide automatic support for the generation of network topologies for large parallel computing systems. This framework is going to be used in the context of the ExaNeSt project to explore large-scale topologies that could be implemented in real systems. The flexibility of both the network to implement routers with different sizes and the framework to optimize different properties make this work very valuable for us.

Our framework considers two of the more popular multi-objective algorithms: NSGA-II and SMS-EMOA. To do that we have developed a new way to represent how the links among routers are established that allows the reduction of the search space and the generation of more regular topologies. We have also developed specific crossover and mutation operators for this particular problem. Moreover, the mutation also includes a mechanism to tune the amount of inter- and intra-cabinet connectivity.

Results for small and large systems show that, for this particular problem, NSGA-II is able to generate consistently better solutions than SMS-EMOA. In addition we have also carried out a collection of experiments showing the performance of applications running in the generated topologies. As baseline we have used an irregular network randomly generated. Results using both single- and multi-path routing policies show the potential of these topologies achieving, in all cases, the best performance.

This work has been a first step towards the development of the framework. We plan to extend it by adding more state-of-the-art multi-objective algorithms such as the recently proposed NSGA-III [11] and new objective functions to allow the generation of different topologies from those of this work. In particular, the framework should be able to generate indirect and hybrid topologies, therefore we will extend it with such topologies. We also plan to report the results of this work for fully populated systems, that due to time restrictions, have not been reported here.

Acknowledgement

This work was carried out within the ExaNeSt project, funded by the European Unions Horizon 2020 research and innovation programme under grant agreement No 671553 and under the EPSRC grants EP/K01568 0/1 and EP/K015699/1 "Interconnection Networks: Practice unites with Theory (INPUT)". Prof. Luján is funded by a Royal Society University Research Fellowship.

References

[1] J. Kim, W. J. Dally, S. Scott, and D. Abts, "Technology-Driven, Highly-Scalable Dragonfly Topology," in *Proceedings of the 35th International Symposium on Computer Architecture*, Washington, 2008, pp. 77–88.

[2] D. Chen, N. A. Eisley, P. Heidelberger, R. M. Senger, Y. Sugawara, S. Kumar, V. Salapura, D. L. Satterfield, B. Steinmacher-Burow, and J. J. Parker, "The IBM Blue Gene/Q Interconnection Network and Message Unit," in *Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis*. New York, NY, USA: ACM, 2011, pp. 1–10.

[3] ORNL, "https://www.olcf.ornl.gov/summit/."

[4] Cray Inc., "http://www.cray.com/assets/pdf/products/xe/idc_948.pdf."

[5] A. Singla, C.-Y. Hong, L. Popa, and P. B. Godfrey, "Jellyfish: Networking Data Centers Randomly," in *Proceedings of the 9th USENIX Conference on Networked Systems Design and Implementation*, ser. NSDI'12. Berkeley, CA, USA: USENIX Association, 2012, pp. 17–17.

[6] M. K. et al., "The ExaNeSt Project: Interconnects, Storage, and Packaging for Exascale Systems," *2016 Euromicro Conference on Digital System Design (DSD)*, vol. 00, pp. 60–67, 2016.

[7] ARM, "https://www.arm.com."

[8] S. Hauck and A. DeHon, *Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2007.

[9] E. Zitzler, M. Laumanns, and L. Thiele, "SPEA2: Improving the Strength Pareto Evolutionary Algorithm for Multiobjective Optimization," in *Evolutionary Methods for Design, Optimisation and Control with Application to Industrial Problems.* Barcelona, Spain: International Center for Numerical Methods in Engineering (CIMNE), 2002, pp. 95–100.

[10] J. Bader and E. Zitzler, "Hype: An Algorithm for Fast Hypervolume-based Many-objective Optimization," *Evolutionary Computation*, vol. 19, no. 1, pp. 45–76, Mar. 2011.

[11] Y. Yuan, H. Xu, and B. Wang, "An Improved NSGA-III Procedure for Evolutionary Many-objective Optimization," in *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*, ser. GECCO '14. New York, NY, USA: ACM, 2014, pp. 661–668.

[12] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II," *IEEE Transactions on Evolutionary Computation*, vol. 6, no. 2, pp. 182–197, 2002.

[13] N. Beume, B. Naujoks, and M. Emmerich, "SMS-EMOA: Multiobjective Selection based on Dominated Hypervolume," *European Journal of Operational Research*, vol. 181, no. 3, pp. 1653 – 1669, 2007.

[14] B. W. Kernighan and S. Lin, "An Efficient Heuristic Procedure for Partitioning Graphs," *The Bell System Technical Journal*, vol. 49, no. 2, pp. 291–307, Feb 1970.

[15] "Analysis of an Equal-Cost Multi-Path Algorithm," RFC 2992, Nov. 2000. [Online]. Available: https://rfc-editor.org/rfc/rfc2992.txt