# Impact of Parallel Workloads on NoC Architecture Design

*Abstract*— **Due to the multi-core processors, the importance of parallel workloads have increased considerably. However, many-core chips demand new interconnection strategies, since traditional crossbars or buses, common for current multi-core processors, have problems related to wires and scalability. For this reason, Networks-on-Chip (NoCs) have been developed in order to support the performance and parallelism focused on several workloads based on message passing. Although a Network-on-Chip is a good option, most designs consist of a large number of routers. These routers are responsible for forwarding packets, and consequently, for supporting message-passing workloads. In this context, the NoC performance is a problem. Therefore, the main goal of this paper is to evaluate the impact of well-known parallel workloads on NoC architecture design. In order to achieve high performance, the results point out to parallel workloads with small packets and cluster-based NoCs with adaptable topologies.**

***NoC Architectures, Parallel Workloads, Performance Evaluation, General-Purpose Many-Core Processors.***

## I. Introduction

Next generations of general-purpose many-core processors point out to exploration of parallel programs in order to achieve high-performance computing [1, 2, 3]. In this context, there are several problems related to different fields and levels, such as: operating systems, algorithms, compilers and architectures. Focusing on architecture, on-chip interconnections are important to support, e.g., collective communication patterns [4]. For this reason, a good choice of network architecture can reduce the packet transmission time in order to increase the performance of parallel programs.

Traditional on-chip interconnection architectures [5, 6, 7] such as buses and crossbar switches have scalability problems. In many-core processors a single and large interconnection reduces the performance due to the wire constraints [8, 9, 10], for instance, resistance and routing. The state-of-the-art points out the Network-on-Chip (NoC) [11, 12] as the main alternative to support a large number of processing cores or on-chip devices.

Networks-on-Chip consist of routers, links and input/output interfaces. Based on this new architecture, message-passing parallel programs [13, 14, 15] can explore packet routing to increase the performance. In this way, collective communications of parallel workloads must be analyzed in order to evaluate a NoC capable of achieving the highest performance. As a consequence, the problem statement is the NoC performance running parallel workloads, which explore different types of collective communication.

In accordance with the problem, the goal of this paper is to present the impact of parallel workloads based on NAS (Numerical Aerodynamic Simulation) Parallel Benchmark (NPB) [15] on NoC architecture design. Therefore, the main contribution is related to the performance verification results of NoC architectures for general-purpose many-core processors and parallel workloads. The results indicate that a cluster-based NoC architecture achieve a higher performance than traditional NoCs based on mesh and torus topologies up to 36% due to the low impact of routers and adaptable topologies.

This paper is organized into the following sections: NoC Architecture Design, Parallel Workloads and Collective Communication, Related Work, Performance Evaluation Methodology, Performance Verification Results, and Conclusions.

## II. NoC Architecture Design

The main component of a Network-on-Chip is the router. It is responsible, among other things, for managing and defining routes, packet flow and quality of service. Typical NoC router architectures are based on specific and non-programmable circuits interconnecting one processing core to the remaining network. This type of router is present in most NoC architectures due to simple design, small area occupation and power consumption.

Figures 1a and 1b show two topologies very common and used in many NoC designs. Mesh and torus topologies have a good diameter, number of links and node interconnections; and there is one router for one core. However, in order to achieve high-performance computing, the number of routers that compose the packet path must be as small as possible.

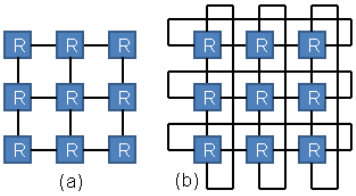

Fig. 1. Traditional NoC topologies. a) 3x3 mesh and (b) 3x3 torus

This work was supported in part by ……

Figure 2 shows a traditional mesh/torus NoC router architecture composed by five input/output ports named North, South, East, West and Node. The packet forwarding technique is based on XY routing; there are input buffers, one arbiter and a simple crossbar switch. This is a non-programmable router that needs an interface unit to convert protocols and transfer packets.

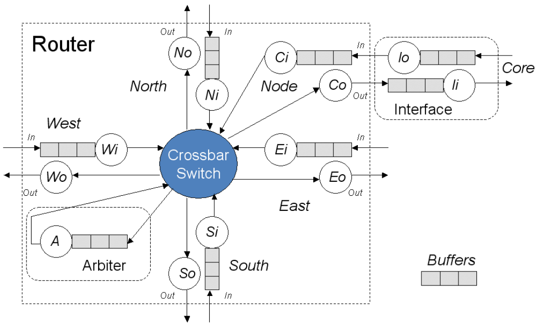

Fig. 2. Mesh/torus NoC router architecture

Another approach is based on programmable routers [16]. Figure 3 shows an architecture that consists of input buffers, a crossbar switch to configure topologies and a Network Processor on Chip (NPoC). This architecture has higher cost, since there is an intra-router processor, but reconfigurable topologies mapped onto crossbar switch (switching node status) increase the performance to forward packets.

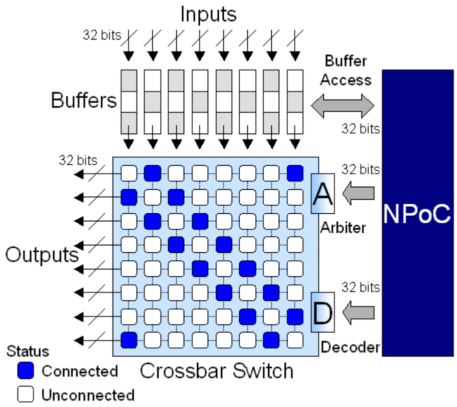

Fig. 3. Programmable router architecture

The router shown in Figure 3 is capable of managing a cluster of cores, and for this reason, it is a good option to cluster-based NoCs [17, 18]. Figure 4 presents a cluster-based programmable NoC [19] that consists of four clusters and a configurable star-based topology. Clusters of cores explore process locality reducing the bottleneck effect of a central router. Due to the crossbar switch presented in Figure 3, it is possible to configure topologies for a cluster of cores or among clusters. The central router in a star topology increases the set of possible configurations, mapping topologies on demand in order to increase the NoC performance. Due to the low number of programmable routers, there is a reduction of hops and the router influence during packet transmissions, besides power consumption [19].

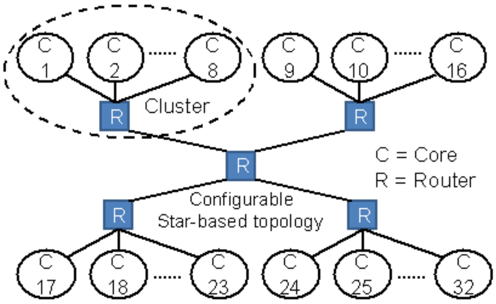

Fig 4. Cluster-based NoC architecture

Therefore, architectures from SoCIN (System-on-Chip Interconnection Network) [20] based on mesh and torus topologies and MCNoC (Multi-Cluster NoC) [19] based on reconfigurable star topology were chosen to be evaluated in this paper. Considerations focusing on power consumption can be found in related work [16, 19].

TABLE I
NOC CHARACTERISTICS

| | SoCIN | MCNoC |
|---|---|---|
| Router latency | 3 cycles | 2 cycles |
| Switching technique | Packet (wormhole) | Circuit |
| Topology | Mesh/torus | Reconfigurable star-based topology |

Table I shows SoCIN and MCNoC characteristics that can influence on final performance results. SoCIN router needs three cycles to read and forward each packet. MCNoC router configures an adaptable topology such as an implemented circuit and forwards the packet in two cycles. As a consequence, MCNoC routers map topologies according to specific communication patterns and SoCIN has a fixed topology to support all patterns.

## III. PARALLEL WORKLOADS AND COLLECTIVE COMMUNICATION

It is important to understand the workload characteristics in order to design NoCs for general-purpose many-core processors. A general or specific behavior can define a design method to achieve a high-performance NoC. In this way, parallel workloads based on collective communications [4, 13, 15] have to be evaluated regarding programming trends for the next generations of many-core processors.

In parallel programming, group of processes can perform an intensive communication by exchanging data packets among them. So, collective communication represents a demand from global data operations that need to be transmitted and received through a message-passing model.

Parallel programming languages are based on functions or primitives to simplify the collective communication description. Figure 5 shows four types of collective communication described as follows: (a) one-to-one: circular shift based on send and receive primitives. (b) one-to-all: based on broadcast where messages from one node are sent to all network nodes. (c) all-to-one: based on gather function different messages from several sources are sent to one node. (d) all-to-all: each node sends a broadcast.

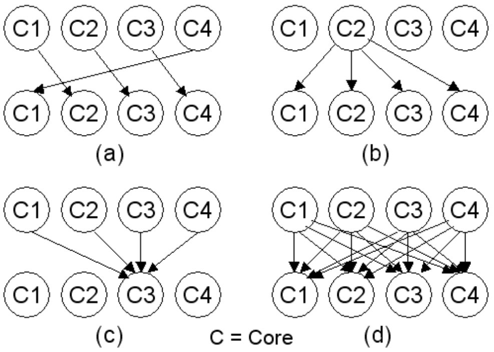

Fig. 5. Collective communication. (a) one-to-one, (b) one-to-all, (c) all-to-one, (d) all-to-all. Adapted from [4]

Focusing on collective communication patterns, this paper describes the impact of parallel workloads in order to investigate the performance of different NoC architectures.

## IV. Related Work

The state-of-the-art shows a large number of research works about performance evaluation of NoCs. Most of the works focus on specific workload behaviors, which do not have characteristics based on message-passing parallel workloads. However, a specific research [21] has focused on parallel workloads for application specific NoCs, and it is the main related work. Therefore, this section describes some evaluation approaches and highlights this main related research.

Traffic generators and analytical models [22, 23, 24, 25, 26, 27, 28] are alternatives to evaluate NoC architectures. Performance results can be obtained from network interfaces and network channels. In addition, synthesis and simulation environments based on hardware description languages are used to validate these models. Performance results are based on specific metrics, such as: injection rate, transferred packets, throughput, and transfer time. The generated traffic and the execution time can be deterministic or random.

Specific programs, such as MPEG-2 encoder, MC-CDMA, routing algorithms, have been evaluated [29, 30, 31] on simulation environments or analytical models. Metrics based on latency, processing time, bandwidth and throughput verify the correct behavior and performance of proposed NoCs.

Application specific NoCs are dedicated architectures for an application domain [21, 29]. For this reason, this type of NoC can achieve a high performance for a special-purpose computing. The main related research [21] describes a design methodology for NoCs based on parallel programs from NAS benchmark to performance evaluation. The goal is to reduce the network contention through the evaluation of well-known communication patterns. Its results show an improvement of performance for generated network topologies related to original mesh topology. The main metric is the relative execution time and the number of blocked messages.

Even though the related research focuses on parallel benchmarks, our contribution described in this paper is different. We focus on impact of collective communication patterns from parallel workloads in order to evaluate the performance of NoC architectures for general-purpose many-core processors. Therefore, our results point out a flexible NoC architecture design that achieves the best performance for a set of parallel programs.

## V. Performance Evaluation Methodology

This section presents the performance evaluation methodology based on three phases: the environment used to collect parallel application traces; the communication patterns (traffic) visualized by _____; and the NoC performance results obtained by the analyzer tool.

### *A. Parallel Application Traces*

Parallel applications based on NAS (Numerical Aerodynamic Simulation) Parallel Benchmark (NPB) [15] version 3.3 were executed on a real eight-core machine. These applications were instrumented to simulate an environment composed of 16 and 32 cores. NAS workloads based on MPI (Message Passing Interface) implementation were compiled with Mpich version 1.2.7p1 using Intel C and Fortran compilers with *O3* optimization. Short descriptions of NAS applications [32] are presented in Table II.

In order to obtain the traces for NAS applications, the Mpich was instrumented using a wrapper to have complete control of each communication procedure. Consequently, all applications were compiled enabling MPE (Multi-Processing Environment) trace option.

Therefore, when a communication primitive is called, the communication wrapper registers information, such as: the name of the MPI operation, start time, the message size, sender and receiver. Just before the end of the communication function, additional information is registered, such as: end time, and execution regions.

After the application execution, a trace file is created with synchronized time to be used by the performance analyzer tool.

### *B. Traffic Visualization by _____*

Visualization tools are widely used during the analysis of distributed applications. They also benefit the behavior understanding of multithreaded applications execution in a single machine with many processing cores. Traditional visualization tools are based on 2 dimensions: one is used to list the resources and other one to show the evolution of

TABLE II
NAS PARALLEL WORKLOADS

| Name | Description |
|---|---|
| BT.A | To solve 3D compressible Navier-Stokes equations with an implicit algorithm. Based on Alternating Direction Implict (ADI) finite differences solver where the resulting system are Block-Tridiagonal, which are solved sequentially along each dimension. |
| CG.A | Conjugate Gradient method used to compute the smallest eigenvalue of a large, sparse, unstructured matrix. Exercising unstructured grid computations and communications. |
| EP.A | Embarrassingly Parallel benchmark, which generates pairs of Gaussian random. Aiming to establish the reference point for peak performance of a given platform. |
| FT.A | Computational kernel of a 3D Fast Fourier Transform (FFT) method. FT performs three 1D FFT, one for each dimension. |
| IS.A | Test Integer Sort operation that is important in particle method codes. This code exercises both integer speed and communication performance. |
| LU.A | Simulated CFD application that uses symmetric successive over-relaxation (SSOR) method to the system resulting from finite-difference discretization of Navier-Stokes equations in 3D by splitting into block Lower and Upper triangular systems. |
| MG.A | Multigrid V-cycle method used to solve the 3D scalar Poisson equation. The algorithm works between coarse and fine grids. It exercises both short and long distance data movement. |
| SP.A | Computational Fluid Dynamics (CFD) application similar to BT. The problem is based on a Beam-Warming approximate factorization that decouples in 3D. The resulting Scalar Pentadiagonal system is solved sequentially along each dimension. |

these resources through time. The limitation of these tools is presented when parallel applications might be influenced by the characteristics of the network topology, or when developers need to understand the communication patterns of parallel applications. In such situations, the analysis using traditional visualization tools is not able to detect performance problems efficiently.

_____ prototype [33] provides an alternative way for analyzing parallel applications. It uses a three dimensional scene to visualize the behavioral evolution of each application component. Two dimensions are reserved to organize these components and the third dimension is used to represent their evolution through time. The direct benefits of this approach are the visualization of communication patterns generated by parallel applications in different time intervals, and the possibility to analyze these patterns together with the network topology involved in the execution.

_____ works by receiving traces that were obtained during the execution of an application. These traces represent the behavior of an application and are composed by events that registered the different functions executed, e.g., point-to-point and collective communication. After providing these traces to the prototype, the user configures a time interval to be analyzed.

Based on this, the prototype extracts all communication and uses this information to generate a graph that represents all communication that occurred within the time interval. By analyzing this graph, the user is able to understand what kind of interconnection can benefit most applications in that time interval.

Figure 6 shows a full communication based on one-to-one collective pattern. Sub-section VI.B shows that in this full communication there are several communication patterns, which can be mapped on adaptable NoC topology.

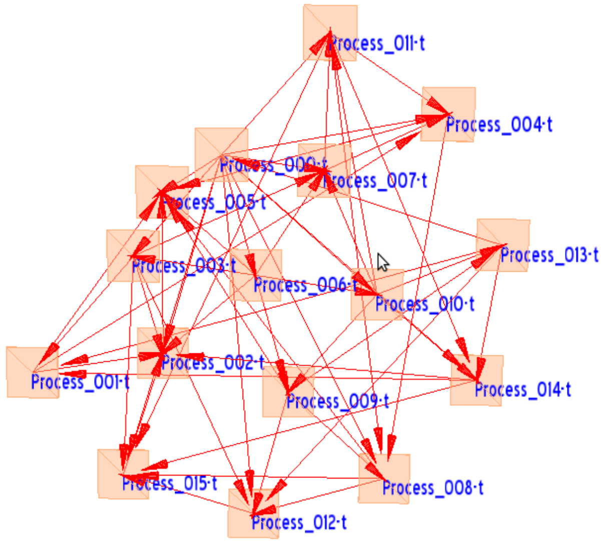

Fig. 6. _____'s visualization of full one-to-one communication (BT trace)

### C. *Performance Analyzer Tool*

As described in Sub-section V.A, an analyzer tool [34] is responsible for reading trace files and processing performance results. This analyzer tool was developed in Python, and it is divided into two main tasks: performance analytical models based on NoC architectures (SoCIN and MCNoC) described in Section II, and transmission time analysis considering time to send packets related to transmission start time. The performance results are specific for each collective communication pattern and for total transmission.

The final or total transmission time (*Tt*) depends, e.g., on network components and on four important metrics that have a direct impact (Equation 1) [35]: latency (*L* - barrier for the first flit to decide the route), bandwidth (*B*), number of packets (*n*) and packet size (*P*). Figure 7 shows that the number of routers can influence directly the total transmission time. For this reason, the analyzer tool is based on a well-known analytical model [35] that expresses the transmission time on each NoC. Both architectures have 32-bit words and a frequency of 100 MHz.

$$Tt = \left( L + \frac{P}{B} \right) * n \quad \text{Equation 1}$$

The router delay (*Rd*) depends on service time (*St* = *Latency*) and buffer time (*Bt*), which depends on the number of packets waiting for processing. For a new packet visit the final delay time is recalculated. A high delay can be considered as a router contention or bottleneck, but for low contentions, the router delay has the same value of service time.

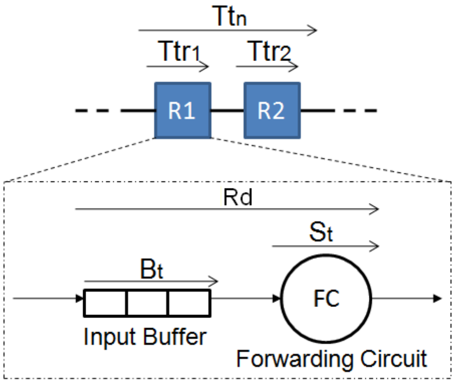

Fig. 7. Performance evaluation metrics

The packet size evaluated in this work has a size ranging from 4 to 128 bytes. All collected messages larger than 128 bytes were divided into 128-byte packets. The analyzer tool verifies only the influence of NoC architectures, considering the relative sum of transmission time to finish a specific parallel application.

Figure 8 shows four examples of summed transmission time considering two metrics: transmission start (*Ts*) and transmission time (*Tt*), collected and calculated from trace file, respectively.

Although all transmission time is evaluated, the final impact is not the total sum of transmissions, but the relative sum of transmissions. For instance, Figure 8a shows an intersection between Tt1 and Tt3 that must be removed, since this does not add time. In the same way, Tt2 does not add time, since Tt1 is larger. Figure 8d shows the sum of transmissions without intersections, thus, each whole time must be considered.

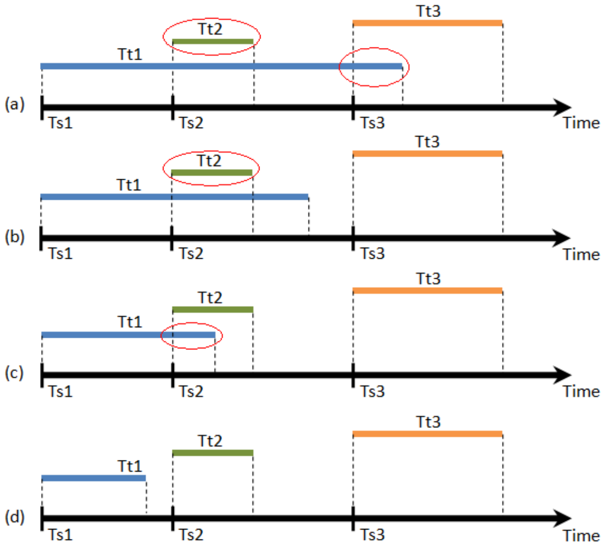

Fig. 8. Relative sum of transmissions: (a) removed transmission and intersection, (b) removed transmission, (c) removed intersection, and (d) no intersection

## VI. Performance Verification Results

This section presents four types of results: i) number and size of collected messages, ii) visualization and analysis based on _____'s screenshots, iii) performance evaluation based on transmission time for each collective communication, and iv) impact on NoC design. The main goal is to verify the impact of each collective communication pattern, visualize time intervals that have specific sub-patterns, and evaluate the transmission performance of NoC architectures.

### A. *Collected Messages*

Table III shows the collected messages from NAS applications considering processors designed to support 8, 16 and 32 processing cores, as described in Section V. Most applications have a high number and large size of messages in the one-to-one pattern, except for EP, FT and IS. They have a low number of transmissions, but especially in all-to-all pattern, FT and IS have large size messages. Although these two applications have large size messages, the number of packets is not so high. For this reason, the impact on NoC architectures is lower than other applications as evaluated in Section VI.C.

Due to the high number of messages, BT, CG, LU, MG and SP applications need efficient NoCs with large packet throughput to reduce the final transmission time. Even though more packets demand more transmission time, the main cost depends on the number of hops of each transmission. For instance, if a pattern has a high number of messages but a low influence of routers, it means a small path or number of hops. Consequently, the NoC architecture has an optimized routing or the application has a clustered communication in neighbor nodes (processing cores).

TABLE III

NUMBER AND SIZE OF COLLECTED MESSAGES

| Patterns | Cores | Messages | BT | CG | EP | FT | IS | LU | MG | SP |
|---|---|---|---|---|---|---|---|---|---|---|
| One-to-one | 8 | Number of messages | 32616 | 23552 | 0 | 0 | 7 | 316338 | 5712 | 65016 |
| | | Maximum size (bytes) | 58080 | 14000 | 0 | 0 | 4 | 81920 | 67600 | 38808 |
| | | Minimum size (bytes) | 9680 | 4 | 0 | 0 | 4 | 128 | 4 | 16000 |
| | 16 | Number of messages | 77280 | 47104 | 0 | 0 | 15 | 759204 | 11024 | 154080 |
| | | Maximum size (bytes) | 34680 | 14000 | 0 | 0 | 4 | 40960 | 67600 | 30720 |
| | | Minimum size (bytes) | 5780 | 4 | 0 | 0 | 4 | 128 | 4 | 9000 |
| | 32 | Number of messages | 260712 | 134656 | 0 | 0 | 31 | 1644936 | 21728 | 519912 |
| | | Maximum size (bytes) | 23360 | 7000 | 0 | 0 | 4 | 40960 | 34320 | 23360 |
| | | Minimum size (bytes) | 2420 | 4 | 0 | 0 | 4 | 64 | 4 | 3240 |
| One-to-all | 8 | Number of messages | 6 | 0 | 0 | 2 | 0 | 9 | 6 | 4 |
| | | Maximum size (bytes) | 12 | 0 | 0 | 12 | 0 | 20 | 32 | 12 |
| | | Minimum size (bytes) | 4 | 0 | 0 | 4 | 0 | 4 | 4 | 4 |
| | 16 | Number of messages | 5 | 0 | 0 | 2 | 0 | 9 | 6 | 3 |
| | | Maximum size (bytes) | 12 | 0 | 0 | 12 | 0 | 20 | 32 | 12 |
| | | Minimum size (bytes) | 4 | 0 | 0 | 4 | 0 | 4 | 4 | 4 |
| | 32 | Number of messages | 6 | 0 | 0 | 2 | 0 | 9 | 6 | 4 |
| | | Maximum size (bytes) | 12 | 0 | 0 | 12 | 0 | 20 | 32 | 12 |
| | | Minimum size (bytes) | 4 | 0 | 0 | 4 | 0 | 4 | 4 | 4 |
| All-to-one | 8 | Number of messages | 2 | 1 | 0 | 6 | 2 | 0 | 1 | 2 |
| | | Maximum size (bytes) | 4 | 4 | 0 | 4 | 4 | 0 | 4 | 4 |
| | | Minimum size (bytes) | 4 | 4 | 0 | 4 | 4 | 0 | 4 | 4 |
| | 16 | Number of messages | 1 | 1 | 0 | 6 | 2 | 0 | 1 | 1 |
| | | Maximum size (bytes) | 4 | 4 | 0 | 4 | 4 | 0 | 4 | 4 |
| | | Minimum size (bytes) | 4 | 4 | 0 | 4 | 4 | 0 | 4 | 4 |
| | 32 | Number of messages | 2 | 1 | 0 | 6 | 2 | 0 | 1 | 2 |
| | | Maximum size (bytes) | 4 | 4 | 0 | 4 | 4 | 0 | 4 | 4 |
| | | Minimum size (bytes) | 4 | 4 | 0 | 4 | 4 | 0 | 4 | 4 |
| All-to-all | 8 | Number of messages | 3 | 0 | 4 | 8 | 33 | 10 | 88 | 3 |
| | | Maximum size (bytes) | 20 | 0 | 40 | 524288 | 530632 | 20 | 16 | 20 |
| | | Minimum size (bytes) | 20 | 0 | 4 | 524288 | 4 | 4 | 4 | 20 |
| | 16 | Number of messages | 2 | 0 | 4 | 8 | 33 | 10 | 88 | 2 |
| | | Maximum size (bytes) | 20 | 0 | 40 | 131072 | 132648 | 20 | 16 | 20 |
| | | Minimum size (bytes) | 20 | 0 | 4 | 131072 | 4 | 4 | 4 | 20 |
| | 32 | Number of messages | 3 | 0 | 4 | 8 | 33 | 10 | 88 | 3 |
| | | Maximum size (bytes) | 20 | 0 | 40 | 32768 | 33488 | 20 | 16 | 20 |
| | | Minimum size (bytes) | 20 | 0 | 4 | 32768 | 4 | 4 | 4 | 20 |

High-throughput NoCs [36, 37] can be presented, e.g., as architectures that map one-to-all pattern better than another one. However, Table III shows low number and small size of messages for this pattern. As a consequence, comparing with one-to-one pattern, one-to-all has a low impact on the final transmission time. But it is important to take into account that all patterns and especially one-to-one have time intervals or transmission periods when occur one or various transmissions. For this reason, NoCs that map topologies according to specific patterns have an advantage, for instance, by supporting sub-patterns from one-to-one pattern as shown in next section (VI.b).

## B. *Visualization and Analysis of Patterns*

Figure 9 shows an interval from BT trace (Figure 6). This interval presents a sub-pattern where it is possible to identify specific communications separated in clusters of cores. For each sub-pattern there is a different behavior that can be expressed as collective communication. For instance, Figure 9a shows one-to-two and one-to-one patterns, and Figures 9b and 9c show one-to-two and two-to-one patterns, and Figure 9d shows two-to-one pattern. Although full one-to-one communication (Figure 6) seems an all-to-all communication, specific intervals can be explored in order to achieve high performance. Consequently, a network-on-chip that supports broadcast can map one-to-three communication better than another that supports just one communication per time slice.

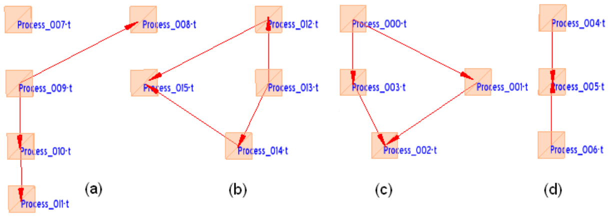

Fig. 9. _____'s visualization of one-to-one interval (BT trace): (a), (b) and (c) two patterns, and (d) one pattern

In the same way, Figure 10 presents other communication interval that can be divided into three interconnected clusters, as follows: Figure 10a shows a star topology that consists of one-to-three and two-to-one patterns. Figure 10b shows three-to-one, two-to-one, and one-to-two communications. These communications can be scheduled in order to explore the performance, such as: three sequences based on one-to-one communication and single one-to-two communication. Figure 10c shows three-to-one and one-to-one communication patterns.

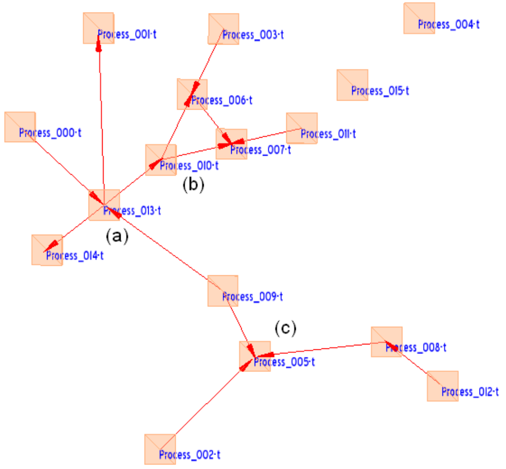

Fig. 10. _____'s visualization of one-to-one interval (BT trace): (a) two patterns, (b) three patterns, and (c) two patterns

Therefore, cluster-based NoCs that support adaptable topologies, e.g. MCNoC, have an advantage in order to map communication patterns or sub-patterns.

### C. *Transmission Time Evaluation*

According to the goal of this paper, this section presents performance evaluations based on transmission time to finish all communication. These evaluations are divided into specific patterns and total transmission time.

Figure 11 shows results based on one-to-one pattern. In accordance with Table III, BT, CG, LU, MG and SP have the highest impact on transmission time. Although LU has higher number and larger size of messages, SP achieves the top of transmission time, performing a highest cost. This occurs since the number of hops is higher and the collective communication holds a larger group of processing cores.

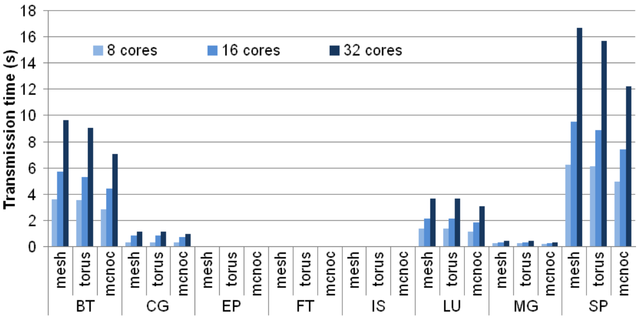

Fig. 11. Transmission time of one-to-one communication pattern

Due to this impact, a Multi-Cluster NoC (MCNoC) achieves a better performance than traditional topologies based on mesh or torus. MCNoC has a lower number of routers, and for this reason, a reduction on performance impact. As presented in Figure 11, MCNoC (32 cores) decreases transmission time relative to 4x8 (32 cores) mesh and torus up to 26% and 22% for BT, 18% for CG, 14% for IS and LU, 23% and 20% for MG, and 26% and 22% for SP, respectively.

Router architectures based on clusters of cores focus on broadcast throughput as an important characteristic. Consequently, for one-to-all pattern MCNoC achieves a highest performance. Figure 12 shows that MCNoC (32 cores) decreases the transmission time relative to 4x8 (32 cores) mesh and torus up to 98% and 97% for BT, 97% and 96% for FT, LU, and MG, and 97% and 95% for SP, respectively. However, the number and size of messages are very low, and for this reason, the impact on the final transmission time is also low. But according to Figure 9 and 10, sub-patterns from one-to-one represent several transmission intervals, and so, architectures that take into advantage on pattern mapping (MCNoC) can explore performance better than traditional interconnections, such as mesh and torus.

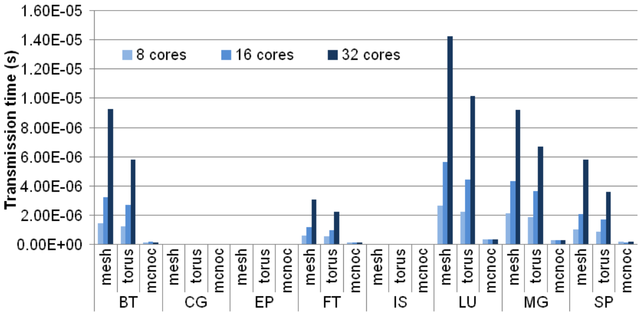

Fig. 12. Transmission time of one-to-all communication pattern

Due to the high competition in all-to-one pattern, the MCNoC performance speedup is not so higher than one-to-all pattern. However, as shown in Figure 13, MCNoC (32 cores) decreases the transmission time up to 25% for all workloads (BT, CG, FT, IS, MG and SP), relative to 4x8 (32 cores) mesh and torus. As described to one-to-all pattern, all-to-one also has a low impact on final transmission time.

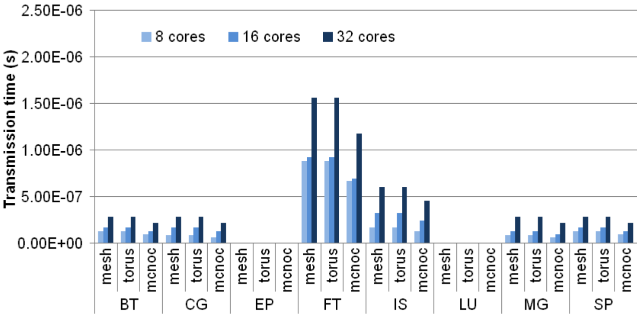

Fig. 13. Transmission time of all-to-one communication pattern

According to Table III, only one application (CG) has no all-to-all pattern. However, the impact of most applications is very low since they have low number and small size of messages. For this reason, Figure 14 highlights FT and IS applications that achieve the highest cost. Although in all-to-all pattern there is a high competition for routers, this type of pattern can consist of two specific operations that perform one-to-all and all-to-one patterns. Consequently, MCNoC explores the broadcast to achieve a higher performance related to traditional topologies than found in all-to-one pattern. Therefore, for all-to-all pattern, MCNoC (32 cores) decreases the transmission time relative to 4x8 (32 cores) mesh and torus up to 33% and 17% for BT, 36% and 21% for EP, 34% and 29% for FT and IS, 35% and 21% for LU, 37% and 22% for MG, and 36% and 20% for SP, respectively.

Figure 14 also shows a different behavior for FT and IS workloads; both have a higher transmission time for NoC versions that support eight cores. The main reason is related to the steady number of messages present for 8, 16 and 32 cores. Due to the reduction of message size from 8 to 32 cores, the transmission time also decreases. According to Table III, this is a typical behavior for FT and IS in all-to-all pattern, other workloads have an increase of messages.

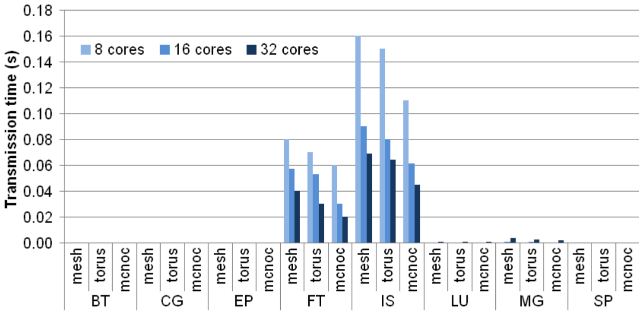

Fig. 14. Transmission time of all-to-all communication pattern

The total transmission time shown in Figure 15 considers the influence of all patterns described in this section. It is important to notice that the main impact on these results is from one-to-one pattern. This is related to the higher number and larger size of messages than other patterns. In addition, the number of hops and the influence of routers add more cost to transmit messages. Therefore, MCNoC (32 cores) decreases transmission time relative to 4x8 (32 cores) mesh and torus up to 26% and 22% for BT, 18% for CG, 36% and 21% for EP, 33% and 28% for FT, 34% and 29% for IS, 14% for LU, 23% and 20% for MG, and 26% and 22% for SP, respectively.

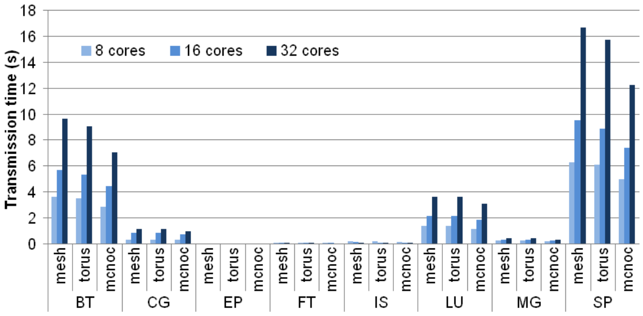

Fig. 15. Total transmission time considering all communication patterns

In order to verify the impact of larger packets than 128 bytes, the same evaluation is summarized in Figure 16 considering the total transmission time for 4096 bytes packets. In this case, there is a reduction of total amount of hops (latency) and, so, a lower influence of routers to transmit all packets. Therefore, the reduction on total transmission time through MCNoC (32 cores) is up to 0.85% and 0.58% for BT, 0.54% for CG, 36% and 21% for EP, 1.13% and 0.8% for FT, 1.49% and 0.94% for IS, 0.19% for LU, 1.22% and 0.89% for MG, and 0.75% and 0.52% for SP, relative to 4x8 (32 cores) mesh and torus respectively. EP workload keeps the same performance described for Figure 15, since it has smaller packets than 128 bytes, as shown in Table III.

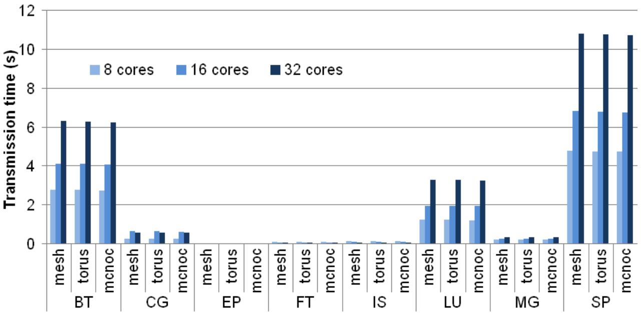

Fig. 16. Total transmission time considering all communication patterns (4096-byte packets)

### D. *Impact on NoC Architecture Design*

Although there is a reduction on transmission time relative to 128-byte packets, it is important to notice that larger packets, e.g., 4096-byte packets, can result in a dedicated time slot for a large period decreasing the packet parallelism and scheduling. For instance, concurrency between two or more parallel workloads, can reduce the throughput for a specific workload, or increase the transmission time, e.g., for EP workload that has small packets, and for this reason, low time slots or link utilization. Therefore, parallel workloads with small packets up to 128 bytes can explore parallelism, throughput and speedup in a cluster-based NoC relative to traditional NoCs based on mesh or torus topologies.

The main problem related to topologies based on mesh and torus is the impact of several router latencies in the packet route. Many packets demand high latencies, but to reduce this impact, output buffers can be alternative instead of input buffers. Through this technique, it is possible to write a new packet in the last position of output buffer as pipeline without stalls that represent latencies. Although output buffers can eliminate latencies, besides the problem of *head of the line blocking*, common for input buffers, an output-buffered crossbar switch needs to be N (number of inputs) times faster than an input-buffered crossbar switch. In this case, there is a competition from all inputs to each output buffer. Moreover, output buffers can demand higher operation frequencies, larger buffers, and consequently, larger area occupation and higher power consumption.

However, there is a probability of frequency reduction by considering a design without optimizations and larger buffers.

Buffers depth is important to define the packet size for parallel workloads. On the other hand, packet size also impact on design decisions of buffers depth. This correlation is necessary to understand parallel workloads for Networks-on-Chip for next generation of many-core processors. For this reason, based on results and characterization shown in Table III, it is possible to estimate buffers depth according to the following evaluation:

- Small packets can explore parallelism and throughput than large packets.
- Lost small packets can reduce the retransmission cost.
- Small buffers are better to circuit switching (evaluated Multi-Cluster NoC).

Therefore, buffers depth between 3 (12 bytes) and 5 (40 bytes) can allocate small packets that are responsible for establishing communication, processing by router, and temporary queuing. Larger packets are received after acknowledgement, but they need to be forwarded through wormhole technique.

Although MCNoC results show the better performance, conventional NoCs based on mesh and torus topologies have advantages of contingency and fault tolerance due to the higher number of routers and links. Effects of faults [38] related to time life can reduce the number of routers, and for this reason, a cluster-based architecture has lower fault tolerance. Besides, in situations of high traffic, more routers and links increase routing options. However, NoC architectures based on mesh or torus topologies able to solve problems related to packet competition are more complex, larger, and consume more power. Output buffers are example to achieve high performance with high cost.

Besides MCNoC advantages presented in this paper, it is important to take into account the flexibility added by programmable routers. If it is necessary to add more functionally to MCNoC, just a new program solves the problem. On the other hand, a conventional NoC based on mesh and torus topologies needs an architecture redesign. Therefore, programmability, topology reconfiguration, and circuit switching are fundamental characteristics that increase MCNoC performance relative to conventional NoC architectures based on mesh or torus topologies.

## VII. Conclusion

In order to achieve high-performance computing, a large number of research works point out many-core processors. As a consequence, networks-on-chip must be efficient with a high throughput and a low transmission time. In this same way, parallel programs must also explore this infrastructure to achieve high performance during execution.

Hence, two concepts can be related: NoC topology and collective communication pattern. NoC topology can be described as fixed or adaptable, based on clusters, or number of interconnected routers. Collective communication is defined by a group of processes that exchanges global data. Therefore, collective communication programs have a high impact on topology of interconnected routers.

Consequently, a cluster-based NoC capable of configuring topologies through programmable routers can achieve a higher performance than a traditional NoC based on mesh or torus topologies. In a cluster-based NoC, there is a lower number of routers, therefore, a lower router impact. In addition, programmable routers can map topologies according communication patterns decreasing the number of hops.

Our results show that the total transmission time running NAS benchmark on MCNoC decreases up to 36% (EP workload) for 128-byte packets. This behavior is present in all programs, highlighting SP workload that has the highest cost, but also a significant reduction on transmission time up to 26%. Strategies to explore high performance and a low transmission time can be related to the number and size of packets (128-byte packets explore a higher MCNoC speedup), but mainly, related to topology and router architectures. Both can increase the performance through flexibility, mapping collective communication patterns and reducing the number of hops.

Future works will focus on impact of parallel workloads running on integrated system based on NoCs and Non-Uniform Cache Architectures.

## References

[1] K. Olukotun, L. Hammond, “The Future of Microprocessors”, ACM Queue, Vol. 3, Issue 7, pp. 26-29, September 2005.

[2] K. Asanovic, et al., “The Landscape of Parallel Computing Research: A View from Berkeley”, Technical Report No. UCB/EECS-2006-183, Electrical Engineering and Computer Sciences, University of California at Berkeley, December 18, 2006.

[3] M. D. Hill and M. R. Marty, “Amdahl’s Law in the Multicore Era”, IEEE Computer Society, pp.33-38, July 2008.

[4] Duato, J., S. Yalamanchili and S. Ni, “Interconnection Networks”, Morgan Kaufmann, 2002.

[5] Agarwal, “Limits on Interconnection Network Performance”, IEEE Transactions on Parallel and Distributed Systems, Vol. 2, No. 4, pp.398-412, October 1991.

[6] R. Kumar, V. Zyuban and D. M. Tullsen, “Interconnections in Multi-core Architectures: Understanding Mechanisms, Overheads and Scaling”, 32nd International Symposium on Computer Architecture (ISCA), pp.408-419, June 2005.

[7] U. Y. Ogras, et al., “On-Chip Communication Architecture Exploration: A Quantitative Evaluation of Point-to-Point, Bus, and Network-on-Chip”, ACM Transactions on Design Automation of Electronics Systems, Vol. 12, No. 3, Article 23, August 2007.

[8] R. Ho, et al., "The Future of Wires", Proceedings of the IEEE, Vol. 89, No. 4, April 2001.

[9] R. W. Keyes, "Fundamental Limits of Silicon Technology", Proceedings of the IEEE, Vol. 89, No. 3, pp.227-239, March, 2001.

[10] R. W. Keyes, "Moore's Law Today", IEEE Circuits and Systems Magazine, Vol. 8, Issue 2, pp.53-54, 2008.

[11] L. Benini and G. D. Micheli, "Network-on-chip architectures and design methods", IEE Proceedings Computers & Digital Techniques, Vol. 152, Issue 2, pp.261-272, 2005.

[12] T. Bjerregaard and S. Mahadevan, "A Survey of Research and Practices of Network-on-Chip", ACM Computing Surveys, Vol. 38, No 1, pp. 1-51, March 2006.

[13] Dongarra, J., et al., "The Sourcebook of Parallel Computing", Morgan Kaufmann, 2003.

[14] Culler, D. E., S. J. Pal and A. Gupta, "Parallel Computer Architecture: a hardware/software approach", Morgan Kaufmann, 1999.

[15] NPB, NAS Parallel Benchmark, http://www.nas.nasa.gov/Resources/Software/npb.html September, 2007.

[16] H. C. Freitas, T. G. S. Santos and P. O. A. Navaux, "Design of Programmable NoC Router Architecture on FPGA for Multi-Cluster NoCs", IET Electronics Letters, Vol. 44, No. 16, pp. 969-971, 31st July 2008.

[17] X. Leng, N. Xu, F. Dong and Z. Zhou, "Implementation and Simulation of a Cluster-based Hierarchical NoC Architecture for Multi-Processor SoC", IEEE International Symposium on Communiations and Information Technology, pp. 1163-1166, 2005.

[18] J. Niemann, M. Porrmann, U. Rückert, "A Scalable Parallel SoC Architecture for Network Processors", IEEE Annual Symposium on VLSI, pp. 311-313, 2005.

[19] H. C. Freitas, T. G. S. Santos and P. O. A. Navaux, "NoC Architecture Design for Multi-Cluster Chips", IEEE International Conference on Field Programmable Logic and Applications (FPL), Heidelberg, Germany, pp. 53-58, 2008.

[20] C. A. Zeferino and A. A. Susin, "SoCIN: A Parametric and Scalable Network-on-Chip", IEEE Symposium on Integrated Circuits and Systems Design (SBCCI), pp. 169-174, 2003.

[21] W. H. Ho, T. M. Pinkston, "A Design Methodology for Efficient Application-Specific On-Chip Interconnects", IEEE Transactions on Parallel and Distributed Systems, Vol. 17, No. 2, pp.174-190, February 2006.

[22] L. Tedesco, et al., "Traffic Generation and Performance Evaluation for Mesh-based NoCs", ACM Symposium on Integrated Circuits and Systems Design (SBCCI), Florianópolis, Brazil, pp.184-189, 2005.

[23] L. Bononi, N. Concer, "Simulation and Analysis of Network on Chip Architectures: Ring, Spidergon and 2D Mesh", Design, Automation and Test in Europe (DATE), 6p., 2006

[24] M. Moadeli, et al., "An Analytical Performance Model for the Spidergon NoC", 21st IEEE International Conference on Advanced Networking and Applications, pp.1014-1021, 2007.

[25] E. Kiasari, et al., "A Markovian Performance Model for Networks-on-Chip", 16th Euromicro Conference on Parallel, Distributed and Network-Based Processing, pp.157-164, 2008.

[26] Z. Lu, M. Zhong, A. Jantsch, "Evaluation of On-Chip Networks Using Deflection Routing", Great Lakes Symposium on VLSI, pp.296-301, 2006.

[27] P. P. Pande, et al., "Performance Evaluation and Design Trade-Offs for Network-on-Chip Interconnect Architectures", IEEE Transactions on Computers, Vol. 54, No. 8, pp. 1025-1040, August 2005.

[28] J. Lemeire, E. Dirkx and W. Colitti, "Modeling the Performance of Communication Schemes on Network Topologies", Parallel Processing Letters, Vol. 18, No. 2, pp.205-220, 2008.

[29] D. Bertozzi, et al., "NoC Synthesis Flow for Customized Domain Specific Multiprocessor Systems-on-Chip", IEEE Transactions on Parallel and Distributed Systems, Vol. 16, No. 2, pp.113-129, February 2005.

[30] R. Lemaire, et al., "Performance Evaluation of a NoC-Based Design for MC-CDMA Telecommunications using NS-2", 16th IEEE International Workshop on Rapid System Prototyping, pp.24-30, 2005.

[31] J. H. Bahn, S. E. Lee, N. Bagherzadeh, "On Design and Analysis of a Feasible Network-on-Chip (NoC) Architecture", IEEE International Conference on Information Technology, pp.1033-1038, 2007.

[32] M. Frumkin, H. Jin and J. Yan, "Implementation of NAS Parallel Benchmarks in High Performance Fortran", In NAS Technical Report NAS-98-009, NASA Ames Research Center, 1998.

[33] ______________________________

[34] ______________________________

[35] Hennessy, J. L., D. A. Patterson, "Computer Architecture A Quantitative Approach", Fourth Edition, Morgan Kaufmann, 2007.

[36] A. Bouhraoua, M. E. Elrabaa, "A High-Throughput Network-on-Chip Architecture for Systems-on-Chip Interconnect", IEEE International Symposium on System-on-Chip, pp. 1-4, 2006.

[37] H. C. Freitas, P. O. A. Navaux, "A High-Throughput Multi-Cluster NoC Architecture", IEEE International Conference on Computational Science and Engineering, pp.56-63, 2008.

[38] A. Aggarwal, et al., "Isolation in commodity multicore processors", IEEE Computer, v.40, n.6, pp.49-59, June, 2007.