Inter-cluster Optimization in Hybrid Interconnect for Heterogeneous Manycore System Designs

***Abstract*—With the increasing demands in data-intensive applications, heterogeneous manycore systems are envisioned which consist of various core clusters and their dedicated NoCs specified in their links and router configurations. It becomes essential for the system designer/ integrator to establish a unified global interconnect network which hides the heterogeneity of intra-cluster data transmission and facilitates global shortcut data transportation for high-performance manycore computing. In this work, we find the optimal RF nodes placement among inter-cluster NoCs based on the application-specific traffic to form a hybrid interconnect infrastructure which is relied upon the recent progress in on-chip RF/wireless interconnect technology. Shortcut channels are established across clusters with high-bandwidth long-range wireless channels. The challenge is how to deploy the RF nodes to achieve global communication efficiency while minimizing the network resource cost. Specially, we minimize the number of shortcut channels while maximizing the per-channel network bandwidth usage. We present an Integer Linear Programming (ILP) formulation for establishing the unified hybrid interconnect network with heterogeneous manycore NoCs, aiming at optimizing the network cost while meeting bandwidth constraints. For large network size problem where the ILP convergence time may take too long, we propose a simple and fast heuristic algorithm which is formulated into a minimum cost maximum flow problem, and a more elaborate heuristic that shows feasible performance when compared to the ILP formulation. The extensive simulation study with real world benchmarks and comparative investigation demonstrate the promising performance while achieving minimal cost resource management.**

I. Introduction

Recent advances in on-chip interconnect technology allow us to integrate not only the massive parallel homogeneous manycore systems, but also heterogeneous manycore architectures on a single chip. They can be utilized for processing many high data-intensive applications, from cloud computing, geosciences, to aerospaces, or bioinformatics. A homogeneous manycore system generally makes it easier for manufactoring, testing, and managing, however, it also lacks of specialization of processing units to the tasks' variation [15]. On the other hand, heterogeneous manycore systems are highly specialized in the application-specific functional units and can be integrated onto the same chip. In addition, heterogeneous manycore systems are the current trend with the integration of large diversity of functional units from different vendors associated with their own configurations as illustrated in Fig. 1. Therefore, the challenge is how to design a global communication platform in order to unify hybrid interconnect among different functional units at minimizing network cost, good performance, and energy-efficient. In this work, we propose the unified global interconnect that consists of the wired and RF/wireless communications among different functional units as clusters. Each cluster can be one or more

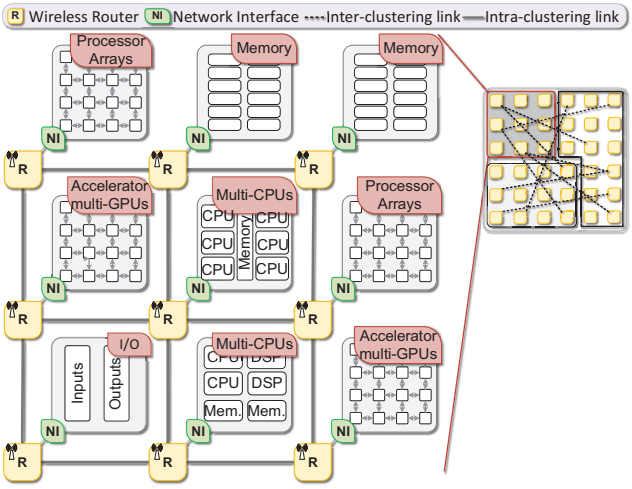


Fig. 1: Illustration of the integrated heterogeneous computational manycore systems platform.

functional units that can communicate with other clusters via RF/wireless communication, as known as inter-cluster communication. The cost of RF/wireless routers placement will be expensively surged if we replace RF/wireless routers to every routers onto the interconnect. Instead of that, we optimize the RF/wireless routers placement for different network topologies based on the application-specific traffic to minimize the network resource cost while maximizing per-link in the global communication. As a result, network throughput and latency are improved whereas power consumption is significantly reduced, when compared to the wired only.

In summary, our contributions of this work includes the following:

- We propose an integrated heterogeneous computational manycore systems platform (in Fig. 1) for various clusters to leverage multi-domain processors that can share the same medium for high data-rate transfers. To overcome the bottleneck with inter-cluster communication, for example, multi-GPUs cross multi-CPUs at high data rate and low-latency, the inter-cluster channels are leveraged for high data-rate transfers among clusters at low-cost (latency) and high performance. These shortcut channels will carry data communication among intermediate clusters to reach their destinations (Section III).
- We exploit the benchmarks and synthetic workloads to propose a fast near-optimal heuristic algorithm (LINCA) to provide feasible solutions of solving heterogeneous manycore system designs problem in polynomial-time. Our detailed comparison shows that LINCA does provide a fast near-optimal to CPLEX in heterogeneous manycore

system designs (Section IV).

- We conduct a rigorous design space exploration in different network layouts, and we see that, there are two types of network topologies including regular and irregular networks that outperform among all possible topologies. Our inter-cluster distribution shows how the quantity of the clusters in the network can be partitioned and its benefits (Section V).
- We feed our optimized solutions from CPLEX Optimizer and the algorithms to the interconnect simulator, GEM5, to justify the network performance corresponding to the optimization results, and to assert the efficiency of our approach in terms of the minimum number of RF/wireless routers placement with low-cost (Section VI).

II. Inter-cluster Communication of Heterogeneous ManyCore Systems

A. Motivation & Rationale

At the system-level design, a heterogeneous manycore system is designed as the integration of a large number of functional/processing units such as RISC/CISC processors, memory, DSP, GPU accelerators, etc., that provide a diversity of services. With the increasing of hundreds or thousands of funtional/processing cores is expected in the near future, it is essential that the on-chip interconnect will be built-in to adapt big data-intensive applications with more bandwith required and greater system management. Thus, the next steps towards exascale computing systems will be the integration of various clusters where each cluster is an individual homogeneous/heterogeneous network-on-chip. Each cluster can be desgined as a general-purpose or specific-purpose to fit the application-domain from different vendors, network topologies to various configurations as in Fig. 1. These clusters will be connected to each other via IP-core and will be joined to the unified global interconnect platform. They can communicate to each other using RF/wireless technologies since the routing layouts among inter-clusters will be extremely complicated and be costly if using bus technologies when scalability is issued.

However, we cannot place all routers to be RF/wireless routers in the heterogeneous manycore system because of three reasons. First, while RF/wireless technologies allow us to replace bus technologies for improving the network performance, the network cost is significantly expensive. Second, we cannot utilize all RF/ wireless routers simultaneously. It depends on data-intensive applications on clusters, only few RF/wireless routers per cluster will be utilized for communicating to other clusters. Third, the interconnect network consumes much more energy for powering all RF/wireless routers while we do not utilize all of them. Hence, the challenge is that how to minimize the number of RF/wireless routers for application-specific traffic while guaranteeing the network performance. As in Fig. 2, we illustrate the mutual relationships between different factors that demonstrate the benefits of using hybrid technology which consists of wired and wireless channels. We see that, in terms of the bandwidth limitation, wireless technology is dominated to the high bandwidth capacity and the network cost, while wired technology has a lower bandwidth limitation and the network cost. This means that, the combination of two technologies yields the good performance in terms of bandwidth limitation, power consumption, and network performance which includes throughput and latency. It also helps reducing the routing layouts among inter-clusters as well as easily managing heterogeneous manycore systems when such systems are scaled up.

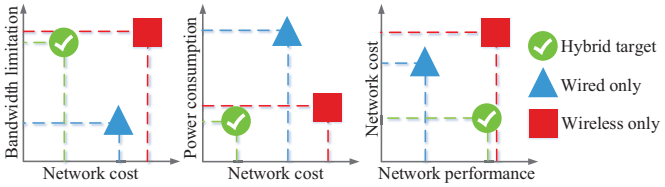


Fig. 2: Impact of hybrid interconnect on network cost versus performance, bandwidth limitation, and power consumption.

Therefore, in this work, we are motivated to optimize the number of RF/wireless routers needed, as well as their optimal placement on clusters corresponding to the application-specific traffic, resulting in the minimum network resource management and improving the network performance in heterogeneous manycore systems.

B. Related work

The authors in [12], [19] proposed the heterogeneous NoC architecture designs with the distribution of resources and various configurations to maximize the benefits in terms of scalability, performance, energy, and QoS. However, these approaches only show the case of heterogeneous configurations onto homogeneous CMPs.

Recent advances of RF/Wireless NoC technology have motivating researchers to propose structured on-chip interconnect network as replacements to buses technologies. The design of these architectures typically satisfies lowering power consumption and latency, good performance. Hybrid interconnect architectures have been proposed to demonstrate the potential benefits from the use of useful features such as wireless backbond infrastructures [9], [16], RF channel modeling [7], [17], [18], [22], voltage frequency island optimization [13], [14] and network scalability [10], [11].

Compared to prior work, our work is based on optimizing RF nodes placement towards the performance efficiency of the unified global interconnect platform in heterogeneous manycore system while minimizing the network resource cost, as well as the quantity of shortcut channels, and maximizing the per-channel network bandwidth usage.

III. Inter-cluster modeling of Heterogeneous ManyCore Systems

In this section, we describe the inter-cluster modeling and how to optimize the shortcut channels for high data-rate communication among inter-cluster while enabling the efficiency of network performance.

For a given connected graph G^k, each $eNOC$ (each cluster) can communicate to other clusters via wired/wireless links by using either intra-cluster or inter-cluster networks. Thus, $G = G^1 \cup G^2 \cup ... \cup G^k = \{V, E\}$, where V is a set of the $eNOC$s ($V \bigcup eNOC^k$), and E is a set of the edges ($E \bigcup E^k$) in network G.

A path p can be established between source cluster and destination cluster to form the communication path, where p belongs to a set of paths $\mathcal{P}_{sd}$, where $\mathcal{P}_{sd} = \{ p_1, p_2,.., p_k \}$. Let α_p be the bandwidth allocation on path p from the source node to the destination node for data transfers. Let $\mathcal{T}_{sd}$ be the traffic from the GPU to the CPU on path p_i in which belongs to a set of traffics $\mathcal{T}$.

To establish the shortcut channels in order to communicate between different inter-clusters in G, let l_n be a shortcut link (a long link). This long link l_n belongs to path p, for instance, from a GPU to a CPU. Let e be a direct communication link (a short link) within intra-cluster. A short link can be applied by PCIe or NVLink [8] to communicate within intra-cluster. Now, we will use long links and short links connected through network topology G. We formulate the network problem as follows.

Let $X = \{X_1, X_2,..., X_n\}$ be a set of $0-1$ variables following the condition as below.

$$X_n = \begin{cases} 1 & \text{if long link } l_n \text{ is established,} \\ 0 & \text{otherwise,} \end{cases} \tag{1}$$

where X_n be a $0-1$ variable that corresponds to long link l_n.

Our objective is to optimize the cost of the integrated heterogeneous network performance, thus, let $C = \{C_1, C_2,..., C_n\}$ be a set of the cost of long links in network G. For each source-destination (S-D) pair, we find all routes in a set of paths $\mathcal{P}_{sd}$ for its transmission. Therefore, the constraints for our problem formulation are formulated as follows.

Each long link is connected between two clusters to transfer data to destination. As in Fig. 3, dashed lines are represented by long links between two clusters and two clusters do not need to be geographically consecutive, as long as they are within transmission range of each other. In this case, a path p_1 is established from traffic pair (S_1, D_1), along with long links l_1 and l_2 (associated with X_{11} and X_{12}, respectively) are connected between cluster 1 and cluster 2, and between cluster 2 and cluster 3, from left to right. In particular, long links l_1 and l_2 are corresponding to each $0-1$ variables X_{11} and X_{12}, respectively. Path p_1 is valid if and only if both X_{11} and X_{12} are ones. Otherwise, it results in the invalidation of path p_1. From this point, we form the constraint for each long link as below.

$$\sum_{\mathcal{T}_{sd} \in \mathcal{T}} \sum_{p \in \mathcal{P}_{sd}} (\alpha_p \times (\prod_{l_n \in p} X_n)) \leq \mathcal{W}_n, \tag{2}$$

where $\mathcal{W}_n$ is the bandwidth capacity of long link l_n. As abovementioned, path p is only valid if multiplication of X_ns equals to 1, thereby, we form $\prod_{l_n \in p} X_n$. This means that it exists the bandwidth allocation α_p on path p, which belongs to a set of paths $\mathcal{P}_{sd}$ from source S to destination D. Then,

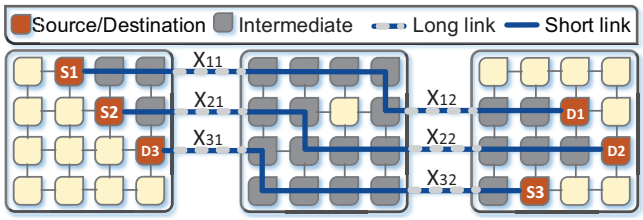


Fig. 3: Illustration of short links (line) and long links (dashed). Short links are connected within intra-cluster. Long links are connected among inter-clusters from source S to destination D in network G.

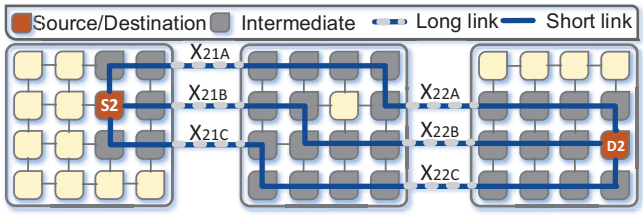


Fig. 4: Illustration of traffic $\mathcal{T}_{sd}$ from source S that goes through inter-cluster to reach destination D in network G.

we find all possible paths from this S-D pair, corresponding to the source's traffic $\mathcal{T}_{sd}$, which belongs to a set of traffic $\mathcal{T}$. Therefore, through the summation of $\mathcal{T}$, and the summation of $\mathcal{P}_{sd}$, we establish this formulation that must not be exceeded the bandwidth capacity $\mathcal{W}_n$ of long link l_n.

As in Fig. 3, each short link e has the amount of traffic that goes through it from different paths of a S-D pair, and e has to belong to the path that goes through long links to reach the destination. Hence, it is similar to the long link's constraint that path p is valid if and only if it belongs to the valid short links and long links. Thus, we formulate the constraint of short link e as follows.

$$\sum_{\mathcal{T}_{sd} \in \mathcal{T}} \sum_{p \in \mathcal{P}_{sd}; e \in p} (\alpha_p \times (\prod_{l_n \in p} X_n)) \leq \mathcal{W}_e, \tag{3}$$

where $\mathcal{W}_e$ is bandwidth capacity on short link e. As mentioned, short link e relies on valid long links to be a valid short link, therefore, it depends on $\prod_{l_n \in p} X_n$. Otherwise, short link e will not be established if multiplication of long links equals to 0. Hence, e is valid following to the long links l_n, corresponding to the bandwidth allocation α_p on path p. Then, through the summation of traffic and the summation of paths from source to destination, the formulation must not be exceeded the bandwidth capacity $\mathcal{W}_e$ of short link e.

In order to guarantee whether or not the bandwidth allocation on each long link l_n on path p from S-D pair is sufficient to meet the source's traffic, it is needed to consider the long link's capacity that is provisioned such that source S's traffic requirement is able to reach destination D as illustrated in Fig. 4. Therefore, we formulate the constraint of the traffic from source S to destination D as follows.

$$\sum_{p \in \mathcal{P}_{sd}} (\alpha_p \times (\prod_{l_n \in p} X_n)) \geq \mathcal{T}_{sd}, \tag{4}$$

where $\mathcal{T}_{sd}$ is the traffic from source S required to send data to destination D. As discussed above, sufficient bandwidth from long link's capacity has to meet the source node's workload requirement in order to ensure that the availability can be met.

From condition (1), and constraints (2), (3), (4) as above-mentioned, we mathematically formulate the inter-cluster modeling for the minimization of the cost of long links in network G as follows.

$$\begin{aligned} \text{minimize:} \quad & \sum_{n=1}^{N} C_n \times X_n \\ \text{subject to:} \quad & (2), (3), (4), \end{aligned}$$

where N is the total number of long links in network G.

IV. Fast Near-Optimal Heuristic Approaches

A limitation of the solvers is that the convergence time may take too long for large network size problem. Therefore, we propose a simple and fast heuristic algorithm, which is formulated into a minimum cost maximum flow problem, that shows feasible performance for solving large network sizes in polynomial-time when compared to CPLEX optimizer. We also implement the greedy algorithm as the baseline comparison in our work.

A. LINCA algorithm

For each application workload, each flow (S-D pair) from application-specific traffic is injected into the network to find a feasible flow with a minimum cost, the cost and bandwidth capacities of the utilized long links are updated to the next iteration for another flow. Our goal is to obtain the minimum cost of long links utilization between clusters.

Consider network graph G with N nodes and M links, including links capacities, traffic, and links' cost. LINCA computes the minimum cost flow for given network flow based on the primal network simplex [6], [20]. Thereby, we can obtain decision variables as long links utilization in network G. Let $G(V, E)$ denote a directed graph, where each link $l(i, j)$ has a maximum bandwidth capacity $A(i, j)$ and associated with its cost $G_c(i, j)$. Let $T(i)$ denote as a traffic in a set of traffic T of an application-specific traffic such as `fmm`, `cholesky`, `radix` from SPLASH2 benchmarks, `UR traffic` from synthetic workload. No links capacity constraints are violated, no traffic flows constraints are violated, and flow conservation is applied at every node in network G.

LINCA computes parameters of network topology G as the inputs, including links capacity A, S-D pairs from traffic T, and links' cost G_c. As in Algorithm 1, LINCA takes the inputs of T, A, and G_c, and iterates through all S-D pairs of a given workload to obtain the output which includes the decision variables for long links utilization. Note that min-cost flow function [6] is partly utilized to achieve our approach. `mincostflow()` function works based on the fast Floyd-Warshall algorithm [3] via dynamic programming. Its time complexity is $\Theta(V^3)$, where V is a set of vertices in a given network. With each additional vertex in each iteration, it has to check every vertex pairs. In this work, we modified min-cost flow function to adapt to our approach.

Algorithm 1 Links Bandwidth Aware Heuristic Algorithm

```
Input:
    traffic T
    Links capacity A
    Cost of links G_c
Output:
    Number of long links utilized & RFs' placement
procedure LINCA (T, A, G_c)
    for each S-D pair do
        Calculate optimal cost using mincostflow()
        Update accumulative links capacity
        Update accumulative decision variables
        Update accumulative cost of links utilized
    end for
end procedure
```

With each iteration for each S-D pair, LINCA provides a feasible solution and then updates the remaining links capacity, decision variables, and the cost of long links utilization in network G for next iterations. LINCA is terminated if there is no more traffic being injected. The output obtained is the minimized number of the utilized long links in the network with given set of traffic.

B. GREEDY algorithm

This section shows the pseudocode of greedy algorithm (as in Algorithm 2) by solving heuristic of making the locally optimal choice at each step with global optimization expected. It locally maximizes the link capacity of the existing communication before establishing new connections. Since it finds locally optimal choices, no guarantee for achieving the global optimum.

V. Experimental Setup

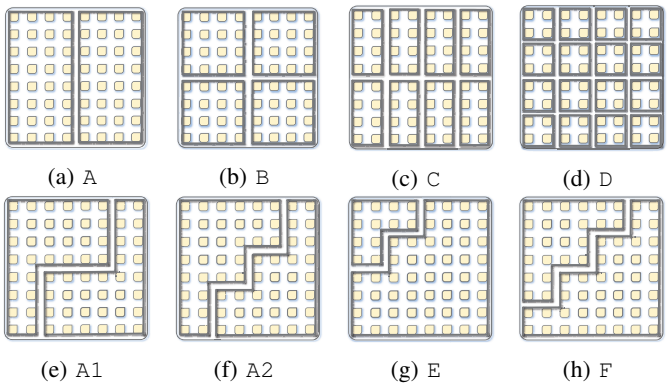
A. Inter-cluster distribution

The baseline of regular (e.g., torus, mesh) and irregular (e.g., various configurations for application-specific traffic) networks in the heterogeneous interconnect (64-node) is configured using different scenarios as in Fig. 5. However, the system designer might choose other types of architectures to fit the expense of their designing complexity and verification. In this section, we propose a design of cluster distribution and links capacity distribution in a given network.

From Fig. 5, we see that the network layout can be partitioned into different ways to implicate the placement of various core clusters with different dedicated NOCs in their configurations, hence, it can be formed as the unified global interconnect network. Hybrid shortcut channels can be established for facilitating global express data transfers for heterogeneous manycore systems. From this point, it depends on the diverse application-specific traffic to find properly RF/ wireless routers placement that can be appropriately placed to minimize the network resource management.

Algorithm 2 GREEDY algorithm

```
procedure GREEDY (T, G)
    Sort traffic T = (T_1, T_2,.., T_n)
    for each T_i in T do
        Find shortest path P_i for T_i in graph G
        if ∃ P_i & remaining capacity CP_i >= T_i then
            CP_i ← CP_i - T_i
            Delete T_i from T
        else
            if ((S_i, D_i)∈ T_i) ⊂ different Clusters then
                if Long Link L(S_i, D_i) ⊄ G then
                    Add L(S_i, D_i) to G
                else
                    for ∀ ∈ Cluster(S_i) & ∈ Cluster(D_i) do
                        Select L_i with highest CP_i & ⊄ G
                            Add L_i to G
                    end for
                end if
            else
                Feasible solution not found
            end if
        end if
    end for
    return Number of long links utilized & RFs placement
end procedure
```



(a) A (b) B (c) C (d) D

(e) A1 (f) A2 (g) E (h) F

Fig. 5: Inter-cluster scenarios on a 8x8 heterogenous many-core network topologies.

B. Simulation Setup

For optimization solvers, we used CPLEX Optimizer [2] to find the optimal solutions, leading to the minimum number of long links utilization in various networks setting. In this work, we used the AMPL language [1] to formulate the equations of the inter-cluster modeling as described in Section III. Then, we used NEOS-server [5] to call the optimizer for finding optimal solutions.

However, when we cope to large network scales, the solvers may take too long to solve the problems in polynomial-time. Hence, for fast design space exploration, the LINCA algorithm was implemented using MATLAB and was used to find a fast near-optimal solution in polynomial-time. GREEDY algorithm was also implemented as a baseline comparison to LINCA and CPLEX.

	Parameters Setting
Processor	ALPHA full-system 64-core, 2Ghz
L1 Cache	32KB 4-way associative, 64B block size
L2 Cache	256KB, 64B block size
Cache Coherence	Two-level directory-based MESI
Main Memory	512MB, 12 cycles control latency, 6 cycles directory latency
Interconnect	Garnet 2D-Mesh with wireless long links
Router configurations	5 cycles routers, crossbar switching technique
Network	16B packet size, 100000 cycles warm-up

TABLE I: Simulation Parameters

Pattern	**Description**
Synthetic	High-radix traffic pattern (uniform random (UR))
SPLASH2 Benchmarks	fmm, lucontig, cholesky, radix

TABLE II: Workload Description

For interconnect network simulation, GEM5 simulator and Garnet [4] were used to feed our optimized solutions from CPLEX Optimizer and the algorithms to verify our inter-cluster modeling in various networks performance. The parameters used in our simulation are summarized in Table I.

We use both synthetic workloads and real world benchmarks for evaluating the impacts of algorithms, topologies, performance, and power consumption analysis. The network is initially warmed up at 100,000 cycles. For synthetic traffic, we use the UR traffic which is categorized as a high-radix workload since a sender node communicates with the large number of sink nodes. For realistic workloads evaluation, four benchmarks listed in Table II were chosen from SPLASH2 benchmarks suite [21]. Traces collected from these benchmarks using GEM5 were used to drive the interconnect network simulator.

To evaluate traffic workloads as in Table II with various networks performance, we setup eight network topologies in a 64-node heterogeneous network for our evaluation as in Fig. 5. These networks are divided into two types, including regular and irregular networks as in Table III. We setup different sets of link capacity ratios among short links and long links to measure the execution of the application traces on these network topologies.

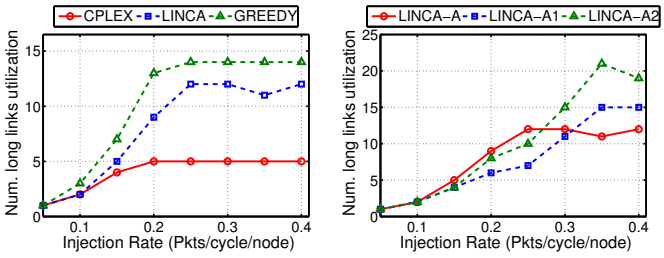
VI. Performance Result & Evaluation

A. Algorithms and Topologies Evaluation

Figs. 6, 7 show the cost comparison of the algorithms and behaviour of various network topologies (S40L80 for synthetic traffic, S1L2 for realistic traffic). In particular, Figs. 6a, 7a show the impact of injection rate/various application-specific traffic on network cost. We see that, CPLEX yields the optimal solution, compared to LINCA and GREEDY. This is because CPLEX solves the problem by making the globally optimal choice for the entire network, the global optimum is guaranteed. In contrast, LINCA and GREEDY

	Description
Regular networks	Types A, B, C, D
Irregular networks	Types A1, A2, E, F
Short links	Bus connections in intra-cluster which are locally connected the nodes within each cluster.
Long links	RF/ wireless channels which are globally connected among clusters via inter-cluster communication.
Sx	Short link associated with x GB capacity.
Ly	Long link associated with y GB capacity.
S40L80	Network configuration with short links capacity is 40GB and long links capacity is 80GB, where $x = 40$ and $y = 80$.

TABLE III: Settings Notation

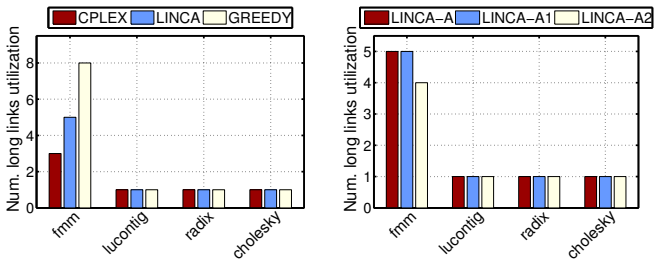


(a) Algorithms (b) Topological changes

Fig. 6: Cost comparison with algorithms (network type A) and topological changes (network types A, A1, A2) with synthetic workloads.

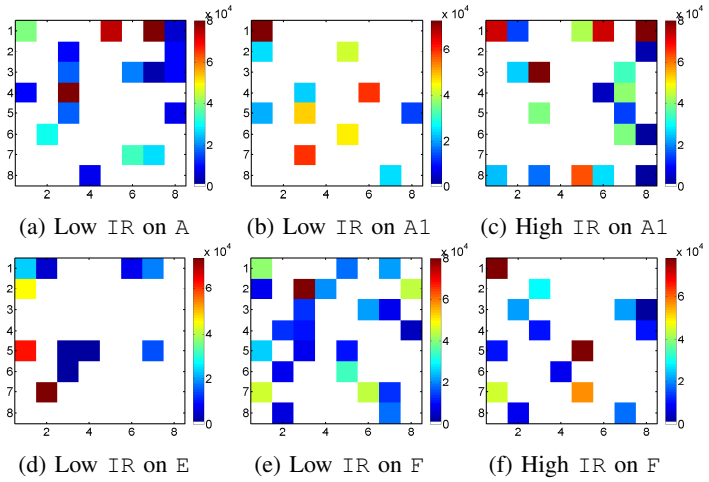
solve the problem by making the locally optimal choice, thus, their results do not guarantee to have the global optimum. In addition, LINCA results are better than that of GREEDY because while LINCA counts on the min cost max flow of the network, GREEDY only relies on the max flow on links to make locally optimal choice.

Fig. 7a shows that fmm application yields a higher network cost when compared to lucontig, radix, cholesky. This is because fmm application (Fast Multipole Method) simulates interactions in two dimensions using different hierarchical N-body method and computes interactions among body cells, then propagates their effects to the bodies



(a) Algorithms (b) Topological changes

Fig. 7: Cost comparison with algorithms (network type A) and topological changes (network types A, A1, A2) with benchmarks workloads.



(a) Low IR on A (b) Low IR on A1 (c) High IR on A1

(d) Low IR on E (e) Low IR on F (f) High IR on F

Fig. 8: Long links utilization and distribution with light UR traffic (IR=0.25) and heavy UR traffic (IR=0.35). Long links utilization is illustrated via spectrum bar, long links distribution is associated with RF nodes placement in different 8x8 network layouts.

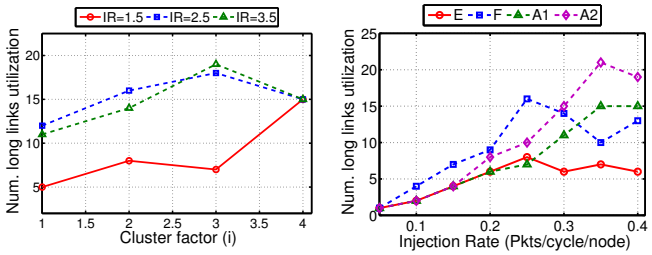
with unstructured communication patterns [21]. Therefore, it generates a heavily application-specific traffic when compared to other traffics. In addition, we observe that, lucontig, radix, cholesky applications have low communication patterns, the network cost for the algorithms is saturated at one long link for inter-cluster communication.

Figs. 6b, 7b show LINCA's behaviour of topological changes on network types A, A1, A2. From Fig. 6b, we observe that, network types A1, A2 are good performance at low injection rate (IR) ($\leq$0.25), but getting worst after IR=0.3. In contrast, network A is stable and good performance after IR=0.3. From Fig. 5, we see that, clusters in network types A1, A2 are much more irregular than that of network A. When traffic patterns become heavy (higher IRs), the capacity on links in the narrowed areas of irregular clusters (in A1, A2) becomes saturated but traffic in these areas still needs to transmit. As a result, more long links are established to preserve the flow conservation law. Therefore, it is extrapolated that network types A1, A2 work better with sparse application-specific traffics, network A works better with dense application-specific traffics.

Fig. 8 shows long links utilized, RF's placement in 8x8 network layouts, as well as the amount of traffic allocation on long links illustrated via spectrum bar based on different topological settings (see Fig. 5). We see that, at low IRs, LINCA algorithm tends to distribute RFs placement around the edges of clusters as in Figs. 8a, 8b, 8d, 8e. When IRs become higher (more dense traffic), RFs distribution is more far-away from the edges as in Figs. 8c, 8f. This is because the algorithm tends to make locally optimal choice in areas with the sufficiency of links capacity for higher IRs.

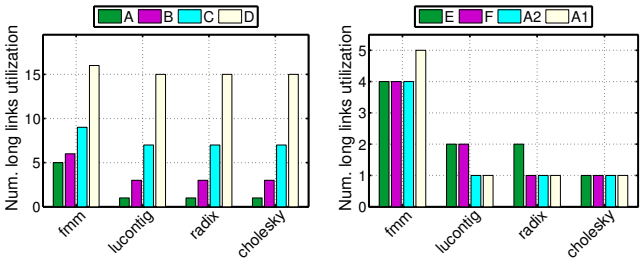
B. Regularity and Irregularity Evaluation

Figs. 9, 10 show the cost comparison of different network settings in various workloads in terms of regularity and irregu-



(a) Regularity (b) Irregularity

Fig. 9: Cost comparison in regular and irregular networks with synthetic workloads.



(a) Regularity (b) Irregularity

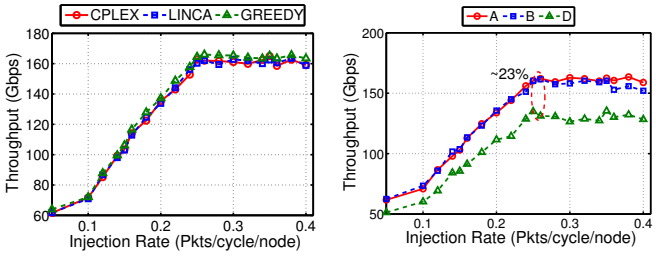
Fig. 10: Cost comparison in regular and irregular networks with benchmarks workloads.

larity. In Fig. 9a, when cluster factor `i` is growing[1] associated with regular network types `A` (`i=1`) →`B` (`i=2`) →`C` (`i=3`) →`D` (`i=4`), we observe that, at low `IRs` ($\leq$`0.15`), the cluster factor `i` should be in the range of (1,3) associated with network types `A, B, C` in order to archieve good performance. At high `IRs` ($\geq$`0.3`), network `D` yields a better performance when compared to other types. Thus, low application-specific traffic is likely suitable for low cluster factor, and high cluster factor yields good performance with high application-specific traffic.

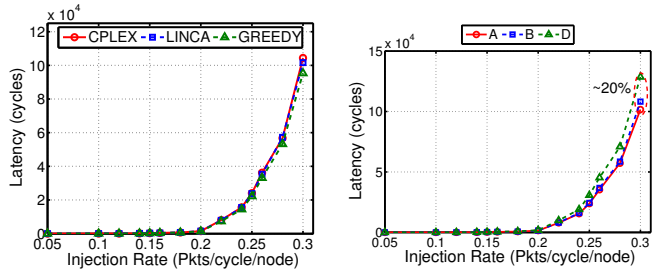
For synthetic traffic, Fig. 9b shows the cost comparison of irregular network types `E, F, A1, A2`. We observe that, network `E` generally yields good performance at both low and high `IRs`. This is because network `E` has sufficient links capacity for high application-specific traffic in areas of clusters (see Fig. 5g).

In addition, network types `A1, A2` yield good performance at low `IRs` ($\leq$`0.30`), however, it does not archieve good performance at high `IRs` ($>$`0.30`) when compared with network `F`. This is because at low `IRs`, narrowed areas in network types `A1, A2` have sufficient link capacity for application traffic, however, at high densitive applications (high `IRs`), these narrowed areas become bottlenecked, as a consequence, more long links established to guarantee traffic conservation, whereas network `F` still have sufficient links capacity for such traffic because the clusters' areas of network `F` are more spreaded-out without narrowed ones.

[1] $Y=\frac{X}{2^i}$, where `i` is cluster factor, `X` is total nodes in network G, `Y` is total nodes in each cluster.



(a) Algorithms (b) Regularity



(c) Algorithms (d) Regularity

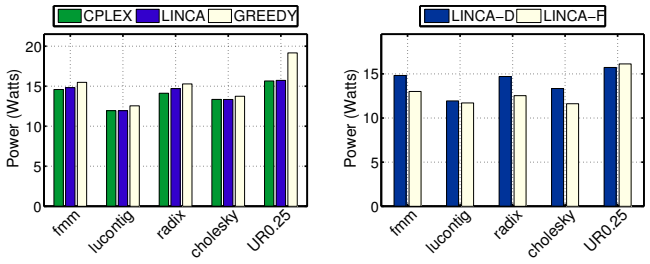
Fig. 11: Network performance with synthetic workloads.

Figs. 10a, 10b show the network cost performance with realistic traffic. In Fig. 10a, regular networks `A, B, C, D` result shows that the network cost is highly proportional to various application-specific traffic. Network `D` gets highest cost because it has highest number of clusters. As a result, more long links needed among clusters to transmit the traffic. At high-radix traffic with `fmm` application in network `D`, one more long link established to guarantee the traffic conservation. For irregular network types `E, F, A1, A2`, these proportional cost still retain since their cluster's areas have sufficient capacity for realistic traffic. Therefore, network cost archived is lower than that of regular ones. In addition, they have only two clusters with various sizes and shapes, the narrowed areas in these types only get bottlenecked when high application-specific traffic is injected (as shown in Fig. 9b).

C. Network Performance Evaluation

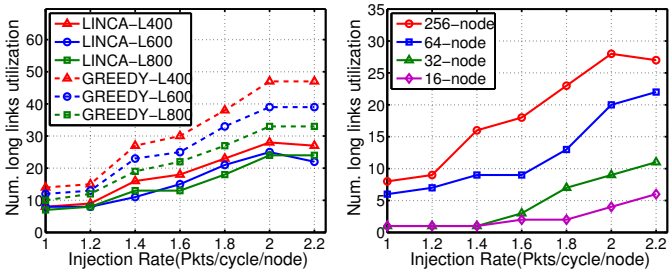
1) Throughput and Latency: Figs. 11a, 11c show that network throughput and latency with the algorithms are approximate in network `A`. This is because the optimized solutions are guaranteed to all bandwidth constraints of the interconnect network, therefore, they yield the similar results in terms of network throughput. This assures that, our approach is feasible in terms of network performance.

Figs. 11b, 11d show that when the number of clusters varies from `2` to `16` clusters (see in Fig. 5), throughput utilization of network `D` becomes lower `23%` of that in networks `A, B`. This is because network `D` has highest number of clusters, and number of long links established among clusters is higher than that of networks `A, B`. As a result, network `D` has more links with high bandwidth but low throughput utilization, since the traffic injections are as the same as other network types. Fig. 11d shows that latency in network `D` is higher `20%` than that of `A, B`. When long links established in network `D` are from one cluster to all clusters, traffic from nearby clusters has to



(a) Algorithms variation (b) Topological variation

Fig. 12: Network power with benchmarks and UR (IR=0.25) traffic.



(a) Ratios (b) Scalability

Fig. 13: Impact of ratios and network scalability.

traverse over the intermediate cluster to reach the destination. In fact, this is a trade-off because if we establish more long links in nearby clusters, the network cost will not be minimal.

2) Power consumption: Power consumption of an on-chip network can take up to 28% of the chip power [19], thus, high performance system design and low-power interconnect are significantly essential for sustaining heterogeneous manycore systems. In this work, we evaluate the behaviors of power consumption in terms of algorithms and topological variations. We suppose that wireless links are higher 10× of power consumption than that of short links. Fig. 12 shows the variation of the algorithms with network D in terms of power consumption. We observe that, network power of LINCA is close to CPLEX, while it is better than GREEDY. This is because the CPLEX yields the lowest number of RF routers, while others produce higher number of RF routers in networks.

3) Ratios & Network Scalability Evaluation: Fig. 13a shows the behaviour of various long links capacities L400, L600, L800 with short link S400 in network A. The result shows the network cost stays retaining proportionally when IRs increased. Fig. 13b shows the scalability of network A in various network size. From this result, we observe that, network size is highly proportional to the input size of traffic.

VII. Conclusion

Hybrid interconnect with shortcut channels overcome the low inter-cluster NOCs bandwidth limitation and establish the unified global interconnect network for manycore computing systems. In this paper, we developed the inter-cluster modeling to quantify the minimal number of long links between clusters while maximizing the per-channel bandwidth utilization and found that RF nodes can be optimally placed to achieve global communication efficiency while minimizing the network resource cost.

Based on this analysis, we developed a fast near-optimal heuristic algorithm, LINCA, that allows a fast approximation of an inter-cluster NOC problem in polynomial-time for fast design space exploration. We presented LINCA using different application-traffic and network topologies, with a comparison of CPLEX Optimizer and GREEDY algorithm as the baselines.

We used the optimized solutions obtained from CPLEX Optimizer and the algorithms to justify the optimal solutions that can meet the communication constraints in the interconnect network via GEM5 simulator with the same inputs (i.e. workloads) we used for the optimizer and the algorithms. Using detailed performance analysis and evaluation, we have demonstrated that RF nodes placement with inter-cluster NOC can be minimized, while guaranteeing the bandwidth constraints, and achieving the maximized per-channel bandwidth in long-range wireless links of the unified global interconnect network for heterogeneous manycore systems.

References

[1] AMPL Optimization. *www.ampl.com*.
[2] CPLEX Optimizer. *www-01.ibm.com/software*.
[3] Floyd-Warshall algorithm. *http://en.wikipedia.org/wiki/Floyd-Warshall_algorithm*.
[4] GEM5 simulator. *http://gem5.org*.
[5] NEOS Server. *www.neos-server.org/neos*.
[6] Network Simplex Algorithm. *https://en.wikipedia.org/wiki/Network-simplex-algorithm*.
[7] A. Brière et al. A dynamically reconfigurable rf noc for many-core. In *25th GLSVLSI*, pages 139–144, 2015.
[8] D. Foley, NVLink, Pascal and Stacked Memory: Feeding the Appetite for Big Data. 2014.
[9] S. Deb et al. Wireless NoC as Interconnection Backbone for Multicore Chips: Promises and Challenges. *IEEE JETCAS*, 2(2):228–239, June 2012.
[10] D. DiTomaso et al. iWISE: Inter-router Wireless Scalable Express Channels for Network-on-Chips (NoCs) Architecture. In *HOTI*, pages 11–18, Aug 2011.
[11] A. Ganguly et al. Scalable Hybrid Wireless Network-on-Chip Architectures for Multicore Systems. *IEEE Trans. Computers*, 60(10):1485–1502, Oct 2011.
[12] B. Grot et al. Kilo-noc: A heterogeneous network-on-chip architecture for scalability and service guarantees. In *38th ISCA*, pages 401–412, June 2011.
[13] W. Jang et al. Voltage and frequency island optimizations for many-core/networks-on-chip designs. In *ICGCS*, pages 217–220, June 2010.
[14] R. G. Kim et al. Wireless noc and dynamic vfi codesign: Energy efficiency without performance penalty. *IEEE TVLSI*, (99):1–14, 1 2016.
[15] M. A. Kinsy and S. Devadas. Algorithms for scheduling task-based applications onto heterogeneous many-core architectures. In *IEEE HPEC*, pages 1–6, Sept 2014.
[16] S.-B. Lee et al. A Scalable Micro Wireless Interconnect Structure for CMPs. In *MobiCom*, pages 217–228, 2009.
[17] D. Matolak et al. Channel modeling for wireless networks-on-chips. *IEEE Comm. Magazine*, 51(6):180–186, June 2013.
[18] H. Matsutani et al. Low-latency wireless 3d nocs via randomized shortcut chips. In *DATE*, pages 1–6, March 2014.
[19] A. K. Mishra et al. A case for heterogeneous on-chip interconnects for CMPs. In *38th ISCA*, pages 389–399, June 2011.
[20] C. Monma and M. Segal. A primal algorithm for finding minimum-cost flows in capacitated networks with applications. *The Bell System Technical Journal*, 61(6):949–968, August 1982.
[21] S. C. Woo et al. The splash-2 programs: characterization and methodological considerations. In *22nd ISCA*, pages 24–36, June 1995.
[22] D. Zhao et al. I(Re)2-WiNoC: Exploring scalable wireless on-chip micronetworks for heterogeneous embedded many-core socs. *Digital Communications and Networks*, 1(1):45–56, 2015.