Tree-Model Based Contention-Aware Task Mapping on Many-Core Networks-on-Chip

Bo Yang\textsuperscript{1}, Liang Guang\textsuperscript{2}, Tero Säntti\textsuperscript{3}, Juha Plosila\textsuperscript{4}
\textsuperscript{1, 2, 3, 4}Department of Information Technology, University of Turku, Finland
\textsuperscript{1}Turku Center for Computer Science, Turku, Finland
\textsuperscript{1}boyan@utu.fi; \textsuperscript{2}liagua@utu.fi; \textsuperscript{3}teansa@utu.fi; \textsuperscript{4}juplos@utu.fi

Abstract— The heterogeneous network-on-chip (NoC) has been proposed as a promising platform for future massive parallel computing. One major design concern on such platform is how to decrease the contention resulted from numerous concurrent communications on the NoC. Contention-aware task mapping is needed in order to alleviate the contention on the network and improve the system performance. In this work, we propose a tree-model task mapping algorithm which takes the bandwidth constraint on the communication channels into consideration. A new model is proposed in the algorithm in order to trade off the weighted communication volume and bandwidth requirement on the NoC. The quantitative comparison shows that the proposed contention-aware algorithm significantly reduces the bandwidth requirement and balances the work load on the communication channels, while the overhead of the weighted communication volume is minimized.

Keywords— Network-on-Chip; Task Mapping; Tree-Model; Contention; Energy Consumption

I. INTRODUCTION

The future system-on-chips (SoCs) will integrate different functionalities to meet diverse requirements of customers. For example, a modern smart phone will support the mobile technology with Internet access, GPS navigation, personal assistant, gaming, and high-speed camera capable of capturing and processing high-resolution images and videos at real-time frame rates. The integration of various functionalities increases the computational complexity of the system. While uni-core processor has reached its limit for high-performance computing, the multi- and many-core processors in which tens to hundreds heterogeneous cores are being embedded, have been proposed to provide greater processing capabilities for future massive parallel computing\textsuperscript{[11]}.\textsuperscript{\textdagger}

With numerous processing cores, the communication amongst cores becomes a major concern for systems based on many-core architecture which has great impact on the performance and cost. In the last decade, NoC has been proposed as a promising alternative for the traditional bus and point-to-point connections in order to address the challenge of increasing concurrent communication requirements as well as the difficulty of global synchronization\textsuperscript{[11]}. A NoC is composed of a set of routers interconnected by communication channels. Each Processing Element (PE) is connected with a router via a network interface to access the communication network. Figure 1 depicts a 4 x 2 2D mesh NoC.

Given a many-core NoC and an application implemented by a set of tasks, the task mapping aims to find a placement on the NoC for each task with the purpose of optimizing design interest, e.g., energy consumption or bandwidth requirement. For the task mapping problem, a few algorithms have been employed in previous works\textsuperscript{[2], [6], [9], [12]}. Based on the energy consumption model presented in [2], most of these works target at minimizing communication energy consumption by mapping two communicating tasks on the NoC nodes as close as possible. This approach works under the assumption that the communication network is congestion-free. Apparently, this could not be the reality. With bandwidth constraint on communication channels, the way of mapping communicating tasks on closest nodes may incur hot-spot in some regions and worsen the congestion on the NoC. In such case, the quality of service will not be guaranteed and the system performance will be degraded.

In this paper we propose a contention-aware task mapping algorithm for many-core NoC. By taking the bandwidth constraint into account at each step of the energy-efficient task mapping, the proposed algorithm aims to minimize the potential contention on the communication channels and reduces the bandwidth requirement of communication. A tree model of a NoC\textsuperscript{[12]} is used for task mapping to improve the efficiency of the mapping algorithm by narrowing down the searching space.

The remaining of this paper is organized as follows. Section II presents related works in literature. The application and NoC models, as well as the formulation of the contention-aware task mapping problem, are given in Section III. Section IV presents the contention-aware task
mapping algorithm. Experimental results are demonstrated in Section V. Finally, Section VI summarizes this paper.

II. RELATED WORK

Hu et al [2] proposed a branch and bound algorithm based on the communication weighted graph (CWG) model. The objective of the mapping algorithm is to minimize the communication energy consumption with constraints of performance handled by bandwidth reservation. Marconet et al [6] proposed an acquaintance dependence model and used simulated annealing algorithm for task mapping. They concluded that compared to CWG, the communication dependence model reduces some application requirements and achieves shorter execution time together with good reduction in the total energy consumption. Srinivasan et al. [9] solved the task mapping problem for mesh topologies using integer linear programming. They claimed that the proposed algorithm achieves higher quality mapping results with lower algorithm complexity. In [7], Murali et al proposed a heuristic algorithm for task mapping on 2D mesh NoCs. The objective is to minimize the average communication delay under bandwidth constraint. They claimed that the bandwidth requirements can be significantly reduced by splitting the traffic among multiple paths.

Using the CWG model, Yang et al [12] proposed an energy-efficient tree-model based task mapping algorithm. A NoC is modeled as a tree composed of a root node and mediate nodes at different levels. By mapping tasks from the root node according to the amount of communication of each task, the algorithm minimizes the communication energy consumption and network delay. In this work, we will extend the tree-model task algorithm by taking the bandwidth constraint into consideration. At each step, the mapping which achieves the minimal weighted communication volume and fulfills the bandwidth constraints on communication channels is preferred.

III. PROBLEM FORMULATION

The inputs of a mapping problem consist of two parts: an application and a many-core NoC. In this work, the application is modeled by a CWG and the many-core NoC is modeled by a computation and communication resources graph (CCRG). For the sake of simplicity, we use a 2D mesh NoC with homogeneous cores as the target NoC, corresponding to the set of CCRG vertices, and T = \{t_1, t_2, ..., t_p\} denotes the set of tiles on the NoC, corresponding to the set of CCRG vertices, and L = \{(t_i, t_j)\} designates the set of communication channels connecting any adjacent tiles \(t_i, t_j\). Each channel \((t_i, t_j)\), denoted by \(l_{ij}\), is tagged with the weight \(bw_{ij}\), representing the bandwidth available on the channel.

The mapping of CWG to CCRG is defined by the one-to-one task-tile mapping function \(map\):

\[
map : V \rightarrow T, map(v_i) = t_i \forall v_i \in V, \exists t_i \in T
\]  (1)

After two tasks \(v_i, v_j\) of an edge \(e_{ij}\) in CWG are mapped, the CCRG, an amount of \(vol_{ij}\) data will be transferred with the bandwidth requirement \(br_{ij}\) along the path from the tile \(map(v_i)(t_i)\) to the tile \(map(v_j)(t_j)\). The path is denoted by \(p_{ij}\).

Definition 3 The Weighted Communication Volume of an Application (WCA) is the sum of products of communication volume \(vol_{ij}\) and the length of the communication path \(|p_{ij}|\) (in hops).

\[
WCA = \sum_{e_{ij} \in E} vol_{ij} \times |p_{ij}|
\]  (2)

Since the energy consumption of a communication \(e_{ij}\) is determined by both the communication volume \(vol_{ij}\) and the length of communication path \(|p_{ij}|\), the objective of the mapping function defined in Equation 1 is to minimize the WCA so that the communication energy consumption is minimized under the bandwidth constraint. The bandwidth constraint is presented as follows:

\[
\sum_{e_{ij} \in E} p_{ij}^{e_{ij}} \leq bw_{st}, \forall e_{ij} \in E, \exists s, t \in \{1, 2, ..., |T|\}
\]  (3)

where the left part of the inequality is the bandwidth occupancy of the communication channel \(l_{st}\), and the \(p_{ij}^{e_{ij}}\) is calculated as follows:

\[
b_{st}^{e_{ij}} = \begin{cases} br_{ij} & \text{if } l_{st} \in p_{ij} \\ 0 & \text{otherwise} \end{cases}
\]

IV. CONTENTION-AWARE TASK MAPPING

The contention-aware task mapping algorithm proposed in this work consists of two stages. At the first stage, the NoC is abstracted into an extended tree. Thereafter, the tasks of the application are sequentially mapped on the tree based on the amount of communication of each task. At this stage, the bandwidth constraint is taken into consideration in order to select the best mapping solution. Figure 2 is an example of mapping an application with 7 tasks on a 3x3 2D mesh NoC using the proposed tree-model based mapping algorithm.

![Fig. 2 Tree-model based task mapping](image-url)
A. Tree Model of a NoC

A NoC is abstracted into a tree by traversing all tiles from the central one. More precisely, the center tile of the NoC is chosen as the root node of the tree, which has most connecting tiles (4 on a 2D mesh NoC). The neighbors of the center tile are abstracted as the child nodes of the root. The procedure repeats until all tiles on the NoC are put onto the tree. The structure is called an extended tree since some child nodes may have more than one parent node. This extended tree presents the tiles and their connections with its neighbor tiles. The example of the abstraction is shown as the first stage in Figure 2.

The level of the extended tree implies some features of the nodes at this level:

1) Resource Quantity:

If we refer to the child nodes of a node as its resources, the nodes close to root (at lower levels) have more resources than those far from root (at higher levels). On the extended tree in Figure 2, the resource quantity of nodes at the root, 1st and 2nd level is 4, 2 and 0 respectively.

2) Communication Distance:

The levels on the tree imply different lengths of communication path from the nodes at a level to the nodes at other levels. For example, in Figure 2, the average communication length (in hops) from any node at the 2nd level to root node is 2, while that from a node at the 1st level is 1.

Due to these two features, it is reasonable for mapping algorithm to map a task on a node at a level as low as possible. By doing this, the newly mapped task gets more child resources with shortest communication distance (1 hop), to map its communicating tasks. Moreover, this task can communicate with those tasks that have already been mapped on the tree through shorter paths. Both of them contribute to the minimization of the WCA defined in Equation 2.

B. Mapping Order of Tasks

After the NoC is abstracted as an extended tree, we can use the extended tree to map tasks. The mapping algorithm takes advantage of the features implied by the level on the tree and starts to map tasks from the root of the extended tree. At the beginning of the mapping, the task having the largest communication volume (CV) with other tasks is selected and mapped on the root of the extended tree. Thereafter, at each step, the task having the largest communication volume with the mapped tasks (CV'), is selected and mapped on the selected objective node which yields the minimal WCA.

C. Objective Node Selection

For each selected task to be mapped, the mapping algorithm needs to find the objective node. All possible nodes on which the selected task is able to be mapped form the searching space. For a mapping algorithm, the searching space determines the complexity of the algorithm. One advantage of using the extended tree is to be able to shrink the searching space. For instance, in Figure 3, we assume that the root and its child nodes have already been used for mapping five tasks, shown as the gray nodes in Figures 3a and 3b. Without using the extended tree, for the selected task, all free nodes on the NoC have to be evaluated in order to find the optimal one. With the extended tree, unnecessary evaluations could be saved.

Note that in Figure 3, the child nodes of the nodes on which the first five tasks have been mapped, denoted as the dotted nodes in Figure 3b, form a circle around their parent nodes (the gray nodes). The mapping on any nodes outside this circle would not be better than those on the circle, with respect to minimizing both the WCA and the bandwidth occupancy of the communication channels. This is because, the mapping on the node outside the circle has to use one additional channel compared with those nodes on the circle. Consequently, the former one will increase the length of communication path by one hop, which results in a larger WCA. In addition, the former mapping would not reduce the bandwidth occupancy on the channels inside the circle. For these two reduce the reasons, to find the objective node, it is reasonable to just evaluate the nodes on the circle, i.e., the child nodes of all mapped nodes on the extended tree (for instance, the nodes on the 3rd level in Figure 3b). This observation is used in this work to shrink the searching space and improve the efficiency of the mapping algorithm.

Fig. 3 Searching space presented by extended tree
D. Bandwidth Constraint

To meet the bandwidth constraint defined in Equation 3, the bandwidth occupancies have to be measured at each mapping step. When a task \( v_i \in V \) is mapped on a node \( t \in T \), the bandwidth occupancy for each channel through which task \( v_i \) communicates with its communicating tasks, has to be updated by adding corresponding bandwidth requirement. As mentioned in Equation 3, the bandwidth occupancy of a communication channel \( l_{st} \) in a partial mapping, \( B_{st}' \), is obtained by the equation:

\[
B_{st}' = \sum_{e_{ij} \in E} b_{st}^{e_{ij}} + \sum_{e_{ij} \in E'} b_{st}^{e_{ji}}
\]

where \( E' \subseteq E \) is the set of edges between those tasks that have already been mapped.

If the \( B_{st}' \) of any communication channel \( l_{st} \) exceeds the bandwidth constraint \( bw_{st} \), the mapping could not be accepted. Then, the mapping heuristic advances to the next node in the searching space. For those mappings which fulfill the bandwidth constraint, we use the following cost function to evaluate them and find the best one.

\[
cost_k = \frac{\sum_{e_{ij} \in E} vol_{ij} \times |p_{ij}|}{\sum_{e_{ij} \in E} vol_{ij} \times NAD} + \frac{\sum_{l_{st} \in L} b_{st}}{\sum_{l_{st} \in L} bw_{st}}
\]

where \( NAD \) is the nodes average distance presented in [5] which is defined as the average distance between two randomly selected nodes on a NoC. For a \( X \times Y \) mesh NoC, the \( NAD \) is:

\[
NAD = \frac{X + Y}{3} \times (1 - \frac{1}{X \times Y})
\]

and the \( L \) is the set of communication channels used in the current mapping. The first fraction in Equation 5 is the measurement of the WCA and the second one that of bandwidth occupancy on communication channels. We use \( cost_k \) to trade off two conflicting interests: minimizing WCA and reducing bandwidth occupancy. The mapping which produces the minimal \( cost_k \) will be accepted as the best one.

E. Contention-Aware Task Mapping Algorithm

Combining the techniques and observations presented previously, the tree-model based contention-aware task mapping algorithm is presented in Algorithm 1. At the beginning of the mapping, the task with the largest CV in CWG is selected and mapped on the root node of the extended tree. Thereafter, the task with the largest \( CV' \) is selected. The algorithm attempts to map the selected task on each node on the circle mentioned in Section IV-C. The \( cost_k \) of current mapping will be calculated and compared with the minimal \( cost_{\text{min}} \) obtained in previous attempts. Finally, the node which achieves the minimal \( cost_{\text{min}} \) will be chosen to map the selected task. The final mapping solution minimizes the WCA and met the bandwidth constraint, which can in turn reduce the communication energy consumption and ease the congestion on the NoC.

Algorithm 1: Tree-Model Based Contention-Aware Task Mapping Algorithm

| Input: | CWG, the extended tree ET of the CRG |
| Output: | Mapping solution \( M \) |

1. Calculate \( CV \) for all tasks \( v_i \in V \);
2. Select the task \( v_h \) with largest \( CV \), map this node onto root node \( t_r \) of ET, create \( V' = \{ v_h \} \) and \( M = \{ t_r, v_h \} \), remove \( v_h \) from \( V \);
3. while \( V \) is not empty do
4. Calculate \( CV' \) for all \( v_k \in V \);
5. Select task \( v_j \) with largest \( CV'_j \) as the task to be mapped;
6. Set the \( cost_{\text{min}} \) with maximum;
7. for each node \( t_i = \text{map}^{-1}(v_i) \), \( v_i \in V' \) do
8. for each child node \( t_c \) of the node \( t_i \) do
9. map task \( v_j \) on node \( t_c \);
10. compute the \( B_{st}' \) of each \( l_{st} \in L' \);
11. if \( B_{st}' > bw_{st} \) then
12. break;
13. else
14. calculate the \( cost_{t_c} \);
15. if \( cost_{t_c} < cost_{\text{min}} \) then
16. \( cost_{\text{min}} = cost_{t_c} \);
17. set the objective node \( t_{obj} \) with the node \( t_c \);
18. Map \( v_j \) onto \( t_{obj} \), append pair \( < t_{obj}, v_j > \) into \( M \);
19. Append \( v_j \) into \( V' \), remove \( v_j \) from \( V \);

V. EXPERIMENT

We evaluated our contention-aware mapping (CA) algorithm against the non-contention-aware (non CA) mapping algorithm in [12]. In this experiment, four real applications were used for the evaluation, including an image processing application (BASIZ) [8], a MPEG encoding application [3], a stereo MP3 decoder application [10] and a H.264 encoder application [4].
For each application, two mappings were obtained using the CA and non CA algorithms. After the mapping, the minimal bandwidth requirement and average bandwidth occupancy on all communication channels are computed under both mappings. Since no bandwidth constraint is set in the non CA mapping, a minimized WCA could be achieved for each application at the cost that a larger bandwidth is required and larger average bandwidth occupancy is resulted on all communication channels. In contrast, a lower bandwidth requirement and bandwidth occupancy are achieved by CA algorithm. Figures 4 and 5 show the normalized minimal bandwidth requirement (BWRE) and average bandwidth occupancy (BW-OC) for the four applications applying CA and non CA algorithm respectively. We can see that for all applications, both the bandwidth requirement and average occupancy are significantly reduced by the CA algorithm. The biggest reduction comes from application BASIZ. The bandwidth requirement and the average occupancy using the non CA algorithm are 17 and 11 times of that of CA algorithm respectively. The smallest increases, which lie in application H.264, are about 5 and 4.5 times of bandwidth requirement and occupancy respectively. Apparently, with a particular bandwidth constraint on a NoC, the mappings produced by the CA algorithm are much better than those of the non CA algorithm with respect to reducing the contention and congestion from numerous concurrent communications on the NoC. Meanwhile, the minimal bandwidth requirement found by the CA algorithm is an important parameter in designing a NoC-based system. A minimized bandwidth requirement can avoid the over-design of a system, which is beneficial both from the design cost and energy saving point of view.

Another beneficial feature of the CA algorithm, as described in Equation 5, is to trade off the bandwidth requirement and WCA of an application running on a NoC. In this experiment, we evaluated a series of bandwidth constraints over the minimal bandwidth requirement for each application. The WCA of mappings using the CA algorithm under such bandwidth constraints were compared. The comparisons show that, with reasonable increase on the bandwidth, the WCA close to the minimized one achieved by the non CA algorithm, could be obtained. Table I shows the WCA gain at the price of increased bandwidth constraint. The second and third column of Table I are the number of nodes in each application and the corresponding NoC size respectively. For application Baseband H.264, a higher increase of bandwidth (24.20% and 16.67% respectively), is required in order to get a relatively smaller WCA. For application MPEG, the overhead of WCA could be within 8.04% if the bandwidth is increased by 6.67%. In the case of application MP3, the smallest overhead of WCA is produced even with the minimal bandwidth requirement. The experiment shows that, the usage of the CA algorithm also provides a method to decrease the overhead of weighted communication volume by tuning the bandwidth constraint within a reasonable range.

### VI. CONCLUSION

A lightweight contention-aware tree-model based task mapping algorithm is presented. By taking the bandwidth constraints into consideration, the proposed algorithm balances the network communication in order to reduce the congestion. Mathematical model which trades off the weighted communication volume and bandwidth requirement is applied in the algorithm. We compared the bandwidth requirements of several real applications using contention-aware and non-contention-aware mapping, and demonstrated that the proposed algorithm significantly reduces the bandwidth requirements (up to 17 times). We also showed that this algorithm can be used to tune the bandwidth constraint to get considerable weighted communication gain which can result in a lower communication energy consumption.

### ACKNOWLEDGMENT

The authors would like to thank the Academy of Finland and Nokia Foundation for the financial support for this work.

### REFERENCES

Liang Guang is a Ph.D. candidate in the lab of computer systems, in the department of information technology from university of Turku, Finland. His research interests include adaptive system design and low-power techniques for parallel embedded system. He has a M.Sc. degree from Royal Institute of Technology (KTH), Sweden.

Tero Säntti is a senior researcher in the Computer Systems Laboratory, in the Department of Information Technology in the University of Turku, Finland. He has a Ph.D. degree from the University of Turku. His research interests include embedded systems, FPGA prototyping and fault tolerant design for space systems.

Juha Plosila is an Associate Professor in Embedded Computing and an Adjunct Professor in Digital Systems Design at the University of Turku (UTU), Finland. He received a PhD degree in Electronics and Communication Technology from UTU in 1999. During 2006-2011 he held a fixed-term post of Academy Research Fellow at the Academy of Finland. Plosila is the leader of the Embedded Computer and Electronic Systems (ECES) research unit and a co-leader of the Resilient IT Infrastructures (RITES) research program at Turku Centre for Computer Science (TUCS). He is an Associate Editor of International Journal of Embedded and Real-Time Communication Systems (IJERTCS, IGI Global). His current research deals with adaptive network-on-chip (NoC) based parallel embedded systems at different abstraction levels, with a special focus on emerging 3D stacked multiprocessor systems. This includes e.g. specification, development, and verification of self-aware, multi-agent monitoring and control architectures for massively parallel 2D/3D NoC systems, as well as applications of autonomous energy-efficient NoC architectures to new computational challenges in the cyber-physical systems domain.

Bo Yang received his M.Sc. degree in Management from the Renmin University of China in 2006. He has carried out several information system projects for five years at China Aerospace Science & Industry Corporation. He has another five years experience in the management field when he served as the head of HR department in aforementioned company. Since September 2008, he has been working in the Computer Systems laboratory, University of Turku as a researcher. He is also a Ph.D. student in the Turku Centre for Computer Science (TUCS), Turku, Finland.

His research interests include power-efficient on-chip application modeling and mapping, reconfigurable network-on-chip platforms, and many-core network-on-chip systems.