Multiple Task Migration in Mesh NoCs Over Virtual Point-to-Point Connections

Lakshmi Prasad.E
PG Scholar,
Arunai College of Engg,
Tiruvannamalai
E-mail: lakshmi_prasad@yahoo.com

Sivasankaran.V
Assistant Professor,
Arunai College of Engg,
Tiruvannamalai
E-mail: pvs.sankaran@gmail.com

Dr.V Nagarajan
Prof/Head –Dept of ECE
Adhiparasakthi Engg College
Melmaruvathur
E-mail:nagarajanece31@rediffmail.com

Abstract - Multiple task migration is a process in network on chips are able to transfer the data from one cluster to another cluster, while transfer the data from one to another cluster message latency, migration latency and power consumption are problem encountered. New techniques virtual point to point Connections (VIPs) has been introduced that dedicates low power and low latency heavy communication flow created by multiple task migration mechanism. The proposed system scheme reduces total message latency, total migration latency, total network latency, power saving is achieved compared to the previously proposed task migration strategy for mesh multicomputer. Analyzing the results show that the proposed scheme reduces message latency by 16% and migration latency by 15%, while 13% power savings can be achieved.

Keywords- *MPSoCs, NoCs, Task migration, Virtual point-to point connection, Power, Performance.*

I. INTRODUCTION

In a mesh multicomputer, submeshes are allocated to perform jobs according to processor allocation schemes, with each task assigned to occupy processors of one submesh with an appropriate size. To assign regions for incoming tasks, task compaction is needed to produce a large contiguous free region. The overhead of task compaction relies mainly on designing an efficient task migration scheme. This paper investigates task migration schemes in 2D wormhole-routed mesh multicomputers with an all-port communication model. Two constraints are given between two submeshes for task migration. Two task migration schemes that follow one of the constraints in 2D mesh multicomputer are then developed. In addition, the proposed schemes are proven to be deadlock-free and congestion-free. Finally, performance analysis is adopted to compare the proposed task migration schemes. The proposed method sets up the virtual point-to-point (VIP) connections over one virtual channel (which bypasses the entire router pipeline) at each physical channel of the NoC. We present two schemes for constructing such VIP circuits. In the first scheme, the circuits are constructed for an application based on its task-graph at design time. The second scheme addresses constructing the connections at runtime using a light-weight setup network. The proposed mechanism is compared to a traditional packet-switched NoC and some modern switching mechanisms and the results show significant reduction in the network power and latency over the other considered NoCs..

II. OBJECTIVES

Runtime task migration was first proposed in multicomputer with load balancing as the major objective. However, specific NoC properties such as limited amount of communication buffers, more sensitivity to implementation complexity, and tight latency and power consumption constraints bring new challenges in using task migration mechanisms in NoCs. As consequence, the efficiency and applicability of traditional migration mechanisms (developed for multicomputers) are under question. Due to the limited resource budget in NoC-based MPSoCs as well as tight performance constraints of running applications, in this paper, we propose an efficient methodology based on virtual point-to point (VIP for short) connections. These dedicated VIP connections provide low-latency and low-power paths for heavy communication flows created by task migration mechanisms.

III. RELATED WORK

Since 1980s, several task migration mechanisms have been proposed for mesh multicomputers [8, 17, 20]. In [8], two migration schemes were introduced. The first scheme, called *Diagonal* scheme, concerns exploring all disjoint paths in one phase to migrate a task based on XY routing. The second scheme, called *Gathering-Routing-Scattering*, reduces the number of task migration paths by collecting the tasks in a limited number of nodes in the source submesh and sends them to the corresponding cores in the destination submesh which will then scatter the tasks to their final destinations. In [20], some constraints are applied for the selection of source and destination submeshes and then two task migration schemes, namely *G-TS* and *NO-TMS*, are proposed for wormhole routing in 2D mesh multicomputer. The constraints mainly concern deadlock-free and congestionfree routing in 2D mesh multicomputer.

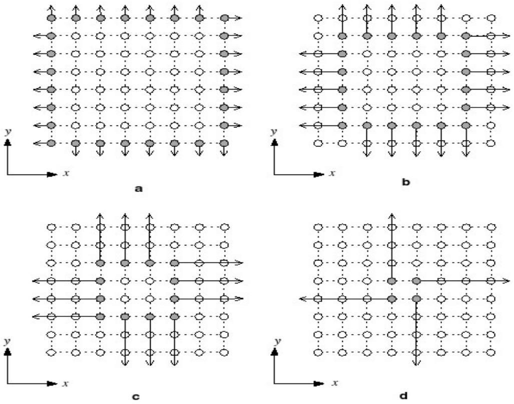

Figure1. Task Migration Process Using G-TMS in an 8 x 8 submesh: (a) first phase; (b) second phase; (c) third phase; (d) fourth phase.

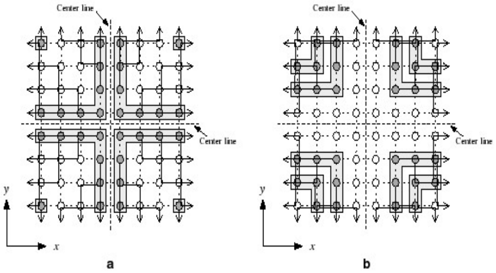

Figure2.Task Migration Process Using NO-TMS in an 8 * 8 submesh: (a) first phase; (b) second phase

IV. PROBLEM FORMULATION

Common approaches in system level design methodology use graph-based description of the studied architecture. An MPSoC with a 2D mesh interconnect can be represented as *Mesh-NoC*=(*S*, *R*, *BW*, *V*, *B*) where *S*(*X*,*Y*) denotes the mesh size (in *X* and *Y* dimensions), *R*(*Src*,*Dest*) defines the routing algorithm that gives the order of intermediate routers met by a message from source *Src* to destination *Dest*, *BW* and *V* represent the maximum bandwidth and virtual channels associated to physical channels and *B* defines the buffer size of input virtual channels2. Task migration between two submeshes *SM*[(X1,Y1),(X2,Y2)] (where (X1,Y1) and (X2,Y2) are respectively the coordinates of bottom-left and top-right corners of the source submesh) and *SM*_[(X3,Y3),(X4,Y4)] (where (X3,Y3) and (X4,Y4) are respectively the coordinates of bottom-left and top-right corners of the destination submesh) in a 2D mesh NoC is defined as movement of tasks of any IP core in *SM* to the corresponding IP core in *SM*_ 2 In the NoC architecture supporting Virtual Point-to-point (VIP) connections [12], all VCs except that for VIP paths have *B*-flit buffers and VIP VCs are equipped with 1-flit latch. An ideal task migration scheme between two submeshes must pose the minimum collision between migration traffic and the normal communication traffic of the application (normal messages have source and destination nodes not residing in the source and destination migration submeshes). To this end, we employ a packed-switched NoC architecture [12] that can provide a number of low-power and low latency dedicated virtual point-to-point (or VIP, for short) connections between any two nodes by bypassing the pipeline of the intermediate routers. As shown in Figure 2 each router in this architecture, uses a multiplexer-tree-based crossbar fabric. In the routers for each physical channel, one virtual channel (VC0) is dedicated to enable bypassing router pipeline stages. This dedicated VC is equipped by a one-flit buffer. The VC0s of intermediate routers are dedicated for constructing VIP connections between certain nodes. The VIP buffer in each router has a signal called *full* and it is set to 1 when a flit enters the buffer to be serviced. Consequently, the flit in the VIP buffer is directed to the crossbar. Otherwise, a VC is selected based on the outcome of routing function, virtual channel allocation and switch allocation. The *full* signal is also used to control the arbiter and allocates the output port to the input port of the VIP in the buffer. Similarly, other buffers in the intermediate routers along the path are connected to the proper output port and establish a VIP connection toward the destination.

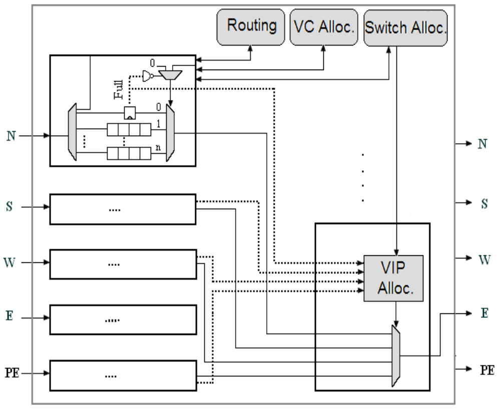

Figure3. Router Architecture [12]

V. PROPOSED APPROACH

The proposed low-power and low-latency task migration strategy is based on *Gathering-Routing-Scattering* algorithm [8] and uses VIPs [12] for the paths involved in task migration to reduce the average migration latency (AML) and hence the total mean network latency and power consumption. In order to reduce the number of task migration paths, a set of IP cores at the source submesh is selected to gather the tasks of the IP cores that are in the same row or column. Similarly, limited IP cores at the destination submesh are used as the intermediate destination nodes of task migration process that further scatter the tasks to their final destination nodes. Since we use dimension order routing (or XY routing algorithm), selected intermediate source and destination cores have to reside in a diagonal arrangement to mitigate possible congestions between different task migration paths. In the first step of task migration process, Gathering step, the task on each core in source submesh SM [(X1,Y1),(X2,Y2)] is gathered in a candidate intermediate core (on the diagonal line) in the same row (if $|X1\text{-}X2|$ _ $|Y1\text{- }Y2|$) or same column (if $|X1\text{-}X2| < |Y1\text{-}Y2|$). Using such rows and columns reduces the number of task migration flows, considerably. The same strategy is used to determine the diagonal cores in the destination submesh. As both source and destination submeshes have the same size, any core of the source submesh corresponds to one core at the destination submesh.

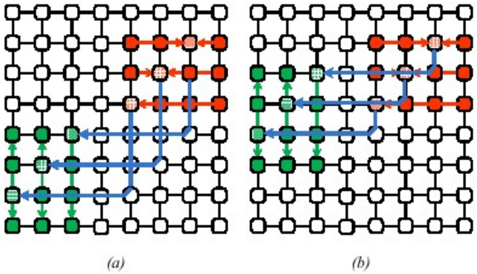

Figure 4: Phases of proposed task migration process when (a) two submeshes do not overlap in X and Y dimension and (b) two submeshes overlap in Y dimension. The Gathering phase is shown in red, the VIP paths are in blue, and Scattering phase is shown in green. Note that hatched cores represent selected diagonal IP cores.

VI. EXPERIMENTALRESULTS

In this section, we evaluate the proposed Multiple task migration schemes using *Gathering-Routing-Scattering* algorithm [8]. All considered multiple task migration strategies and the proposed technique are implemented on an NoC architecture simulated by Xmulator [2]. Xmulator [2] is a fully-parameterized discrete event simulator for interconnection networks which is augmented with Orion library [19] to calculate the power consumption of the network. Also, the hardware and software requirements for VIP establishment and control networks are emulated in this environment. Simulation experiments are performed for a 128-bit wide system. Moreover, the process feature size and working frequency of the NoC is set to 65nm and 280 MHz,

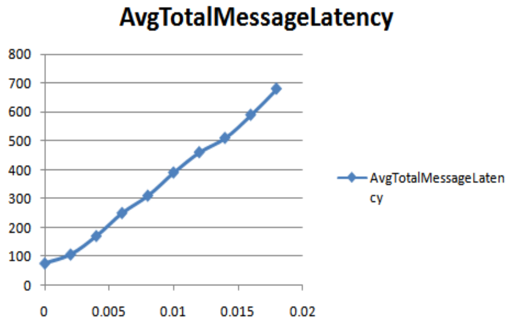

Figure 5: Average Message Latency

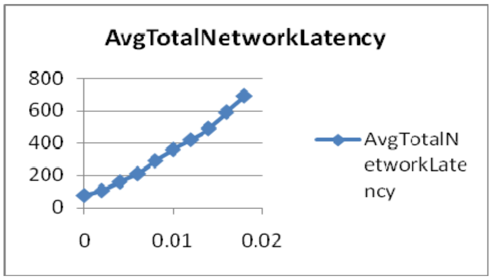

Figure 6: Average Total Network Latency

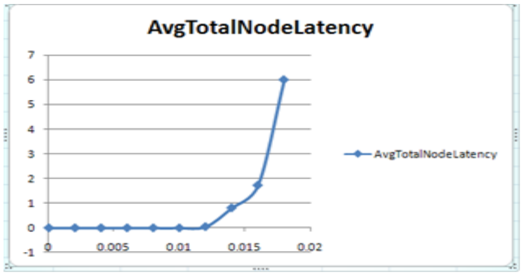

Figure 7: Average Total Node Latency

In summary, the proposed VIP-based scheme has reduced the average message latency by 16%, average node latency by 15%, and total network power by 13% with respect to *Gathering-Routing-Scattering* method.

VII. CONCLUSION

In this paper, we proposed Multiple task migration scheme in mesh-based NoCs based on low-latency and low-power virtual point-to-point(VIP) connections. Experimental results revealed that the proposed scheme could improve over earlier technology. This improvement was due to a series of strategies used in the proposed method i.e. multiple task migration was enhanced the message generation rate with in short span of latency during VIP establishment, migration VIPs were prioritized over normal VIPs. Since a new VIP connection could tear down previously established VIPs (if suitable), we could even prioritize scenarios with multiple task migration and hence network performance improvement can be achieved. Moreover, the VIP paths were limited to connections between some selected diagonal cores in the source and destination submeshes. This can better help reducing stalled normal message flows during VIPs.

REFERENCES

1. Aguiar, S.J. Filho, T.G. dos Samtos, C. Marcon, and F. Hessel, “Architectural support for task migration concerning MPSoC,” in *Proc. WSO Workshop*, *SBC*, 2008, pp. 169-178.
2. L. Benini, and G. De Micheli, “Networks on chip: a new paradigm for systems on chip design”, *IEEE Computer*, Vol. 35, 2001, pp.70-78.
3. S. Bertozzi, A. Acquaviva, D. Bertozzi, and A. Poggiali, “Supporting task migration in multi-processor systems-on-chip: a feasibility study,” in *Proc. DATE*, 2006, pp. 15-20.
4. E.W. Briao, D. Barcelos, F. Wronski, and F.R. Wagner, “Impact of task migration in NoC-based MPSoC for soft real-time applications,” in *Proc. VLSI*, 2007.
5. E.W. Briao, D. Barcelos, and F.R. Wagner, “Dynamic task allocation strategies in MPSoC for soft real-time applications,” in Proc. *DATE*, 2008, pp. 1386-1389.
6. E. Carvalho, N. Calazans, and F. Moraes, “Congestion-aware task mapping in NoC-based MPSoCs with dynamic workload,” in *Proc. ISVLSI*, 2007, pp. 459-460.
7. T.S. Chen, “Task migration in 2D wormhole-routed mesh multicomputers,” *Information Processing Letter*, no. 73, pp. 103-110, 2006.
8. L.K. Goh and B. Veeravalli, “Design and performance evaluation of combined first-flit task allocation and migration strategies in mesh multicomputer systems,” *Parallel Computing*, vol. 34, no. 9, pp. 508- 520, 2008.
9. J. Hu and R. Marculescu, “Energy- and performance- mapping for regular NoC architectures,” *IEEE Transactions on Computer Aided- Design and Integrated Systems*, vol. 24, no. 4, pp. 551-562, 2005.
10. H. G. Lee, et al., “On-chip communication architecture exploration: a quantitative evaluation of point-to-point, bus, and Network-on-Chip approaches”, *ACM Transactions on Design Automation of Electronic Systems*, Vol.12, No.3, 2007.
11. M. Modarressi, H. Sarbazi-Azad, A. Tavakol, “Virtual point-to-point connections for NoCs,” in *Proc. NoCs*, 2009, pp. 203-212.

12. S. Murali and G. De Micheli, “Bandwidth-constraint mapping of cores onto NoC architectures,” in *Proc. DATE*, 2004, pp.896-901.
13. V. Nollet, T. Marescaux, P. Avasare, D. Verkest, and J.Y. Mignolet, “Centralized run-time resource management in a network-on-chip containing reconfigurable hardware tiles,” in *Proc. DATE*, 2005, pp. 234-239.
14. V. Nollet, P. Avasare, J.Y. Mignolet, and D. Verkest, “Low cost task migration initiation in a heterogeneous MP-SoC,” in *Proc. DATE*, 2005, pp. 252-253.
15. O. Ozturk, M. Kandemir, S.W. Son, and M. Karakoy, “Selective code/data migration for reducing communication energy in embedded MpSoC architectures,” in *Proc. GLSVLSI*, 2006, pp. 386-391.
16. T. Streichert, C. Strengert, C. Haubelt, and J. Teich, “Dynamic task binding for hardware/software reconfigurable networks,” in *Proc. SBCCI*, 2006, pp. 38-43.
17. R. Tornero, V. Sterrantino, M. Palesi, and J.M. Orduna, “Multiobjective strategy for concurrent mapping and routing in networks on chip,” in *Proc. IPDPS*, 2009. pp. 1-8.
18. H. S. Wang, X. Zhu, L. S. Peh, and S. Malik, “Orion: a power performance simulator for interconnection networks,” in *Proc. Int. Symp. Microarchitecture*, Nov. 2002, pp. 294-305.
19. N.C. Wang and T.S. Chen, “Task migration in all-port wormhole routed 2D mesh multicomputers,” *Information Sciences*, no. 176, pp. 3409-3425, 2006.
20. W. C. Woo, M. Ohara, E. Torrie, J. Pal Singh, and A. Gupta, “The SPLASH-2 Programs: Characterization and Methodological Considerations,” in *Proc. ISCA*, Jun. 1995, pp. 24-36.
21. J. Wooyoung and D.Z. Pan, “A3MAP: architecture-aware analytic mapping for networks-on-chip,” in *Proc. ASP-DAC*, 2010, pp. 523- 528.a