TELKOMNIKA, Vol. 11, No. 7, July 2013, pp. 3546 ~ 3551
e-ISSN: 2087-278X

Intra-Inter Triplet Object Interaction Mechanism in Triplet-Based Hierarchical Interconnection Network

Talpur Shahnawaz[*1,2], Qureshi Imran Ali[2], Khahro Shahnawaz Farhan[1], Kavita Tabbassum[3], Erum Saba[3]
[1]Beijing Institute of Technology, Beijing, P. R. China
[2]Mehran University of engineering and Technology Jamshoro, Sindh, Pakistan
[3]Information Technology Center, Sindh Agriculture University Tandojam, Pakistan
*Corresponding author, e-mail: talpur@bit.edu.cn[*1], shahnawazfarhan@gmail.com

Abstract

Object oriented languages usually avoid direct message passing, due to its complicated implementation, though that is the promising way to communicate in concurrently inherited objects. With the advancement in the high performance computing system, interaction between parallel application objects onto physical cores becomes one of the significant issues, which is not fully explored yet. In object oriented programming attribute data is included in objects and their state can be changed using the methods. Objects enable massage passing to other objects interacting with each other. Comprehensive problems can be molded by object-oriented methodology, and solves difficult program running object-oriented programs. Cores communicate with each other through communicator and groups in MPI, but in our reference architecture TBHIN (Triplet Based Hierarchical Interconnection Network), the cores are already faction in Triplets. We propose IITOIM Model to improve the performance with efficient intra-inter triplet cores communication mechanism between the objects in TBHIN.

Keywords*: TBHIN, CMP, OOP, Message Passing, Multi-Core*

1. Introduction

A lot of research is being done on chip multi processor and high performance computing. However multiple cores are incorporated on chip to increase the efficiency of computation. Interconnection scalability, flexibility and reliability can be achieved when Network-on-chip enables more integrated cores on system-on-Chip.

Quite a lot of application are successfully developed in high performance computing with high degree of thread parallelism . Common to these applications is MPI (message Passing interface) framework and the platform is cluster computing with hundreds of CPUs but the researchers also mention some demerits of MPI in embedded multicore systems as the library of MPI is a big obstacle in accommodating in local memory.

In current domain of computing, the focus of the venders is CMP architecture with growing number of cores. The application should be designed in such a manner that can easily be adopted by processors to run them in multithread. Object oriented standard allows code maintenance, reusability and scalability [1]. Multiple objects accumulate a class in object oriented approach. Each object is indentified as solo entity [2]. For decades object oriented programming is the mainstream for developing application but objects on physical cores in HPC need more attention to map parallel programs [3]. Message is the common way through which objects can interact with each other.

In the multicore era there are the object oriented architecture which supports objects directly mapped on the cores. Few significantly 2DMesh multicore architectures, base on three interconnect structure with hierarchical grouping of shared storage including tile layout and multi-modal new technology features of CMP. Those have hardware support on the message mechanism and object management.

The development process of object-oriented software finds such a wide range of applications, due to inherent parallelism [4][5][6][7]. The main force lies in the emergence of object-oriented technology, significantly reducing software development effort. Based on the research of multicore message communication paradigm, in this paper we propose object interaction/communication Mechanism within a single triplet core and cores outside that triplet called Intra-Inter-Core Object Interaction mechanism (IITOIM).

Received January 10, 2013; Revised March 13, 2013; Accepted March 30, 2013

The paper is organized with different Sections. Section 2 illustrate object hypothesis. Section 3 demonstrates communication model. Section 4 expresses mechanism and evaluation. In Section 5 related work is discussed. Paper will be concluded in section 6.

2. Object Messaging Hypothesis

2.1 Message necessity between objects

The methods and data build an entity, called object. Generally multiple objects employ a significant role in message paradigm particularly interacting with each other. Different object oriented languages provide different mechanisms for interacting between objects like in java by using parameters.

In object oriented programming, methods are called either directly or by passing parameters. The same procedure is known as message passing when the methods exist on one core and are called by objects running in an another core in multicore architecture. The message send and receive is additional course of action comparing to the conventional methods calling in OOP.

Some researcher thought there should be some information stay alive, when the object at one core receives message from sender object of other core. In other words the message handler will be called for appropriate action. The interaction between the objects could directly or indirectly depend on the send and receive initiator and the synchronous and asynchronous communication.

2.2 Object Association on cores

There are multiple objects running on each core as shown in Figure 1. Object consists of data and functions operate on that data. Each object is the owner of an individual queue of messages that stores every service requested to it [8]. Each core is also owner of object queue within the Triplet. There will be an object table for object recognition among each other, apart from intra or inter core communication. Our model is different from MPI. We don't need grouping which is extra burden on processor to rank the core in each group. In our scheme within a Triplet interaction between objects is supposed "intra core" and between the Triplets is said "inter core" communication.

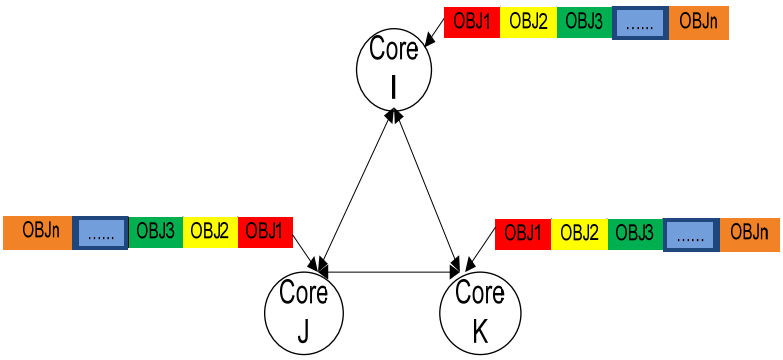

Figure 1. Inter Core Interaction

3. Object Communication Model in TBHIN

3.1 An Intra Triplet Object communication

In Figure 1, assume that when first object of core I, likes to communicate to core J's second object, the message is created to send OBJ2 of J, which receive the message and send back the handler address. OBJ1 of I receive the message and execute the method required or may need data to complete the job. In the TBHIN Object oriented architectures, it will be decided according to triplet code whether the message is for the object in the same triplet or for the object to different triplet according DDR algorithm [9][10].

3.2 An Inter Triplet Object communication

We suppose that core I's first object interacts with the Core Q's first object. First it is decided according to the triplet code that the message is not for the object of same triplet. According to the routing scheme in TBHIN the message transfers from source core to target core and then to the target object.

Three triplets with nine cores are shown in Figure 2. For identification of each object on the core in the triplet, we build object identification with the Core ID expressed in [2]. The object ID is created as the object is entered in object queue with FIFO fashion. The sequence number will be generated for that object e.g Sq1, Sq2, …. Sqn. Every core is connected in Triplet-Based Hierarchical Interconnection Network faction as described in the addressing as given in Table 1. According to [4] the sequence number (Sq) will be added to generate object ID.

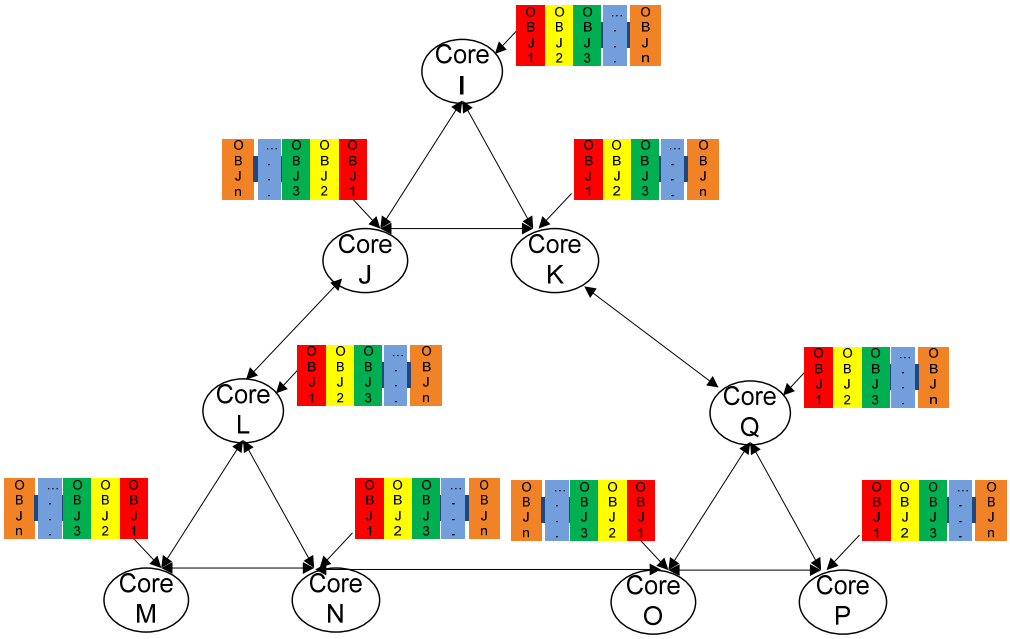

Figure 2. Intra/Inter Triplet Objects Communication Machenism in TBHIN

Table 1. TBHIN Adressesing

Triplet 1(01)			Triplet 2(10)			Triplet 3(11)		
Core 1	Core 2	Core 3	Core 4	Core 5	Core 6	Core 7	Core 8	Core 9
0101	0110	0111	1001	1010	1011	1101	1110	1111

3.3 Unique object Sequence Number

Every core is recognized with alphabet and the object is distinguished with alpha numeric label "obj" plus number according to the entrance in queue, making Unique Object Sequence (UOS) as identification. The objects will be acknowledged through UOS in sending and receiving messages between source core and target core.

Each object may have several threads. The ID for threads will be generated "obj+number+Tnumber". The thread queue will be maintained in each object at L1 cache as already expressed in [8] where each object has individual queue.

4. Proposed Mechanism

Triplet Based Hierarchical Interconnection Network object oriented architecture has processing unit performs the same operation as ALU in general CPU ,except the I/O Operation which is supported by message queue. Objects are modified by methods running in processing part and data is stored in Data part L1 cache and so on. Processing part send messages to other objects through interconnected part.

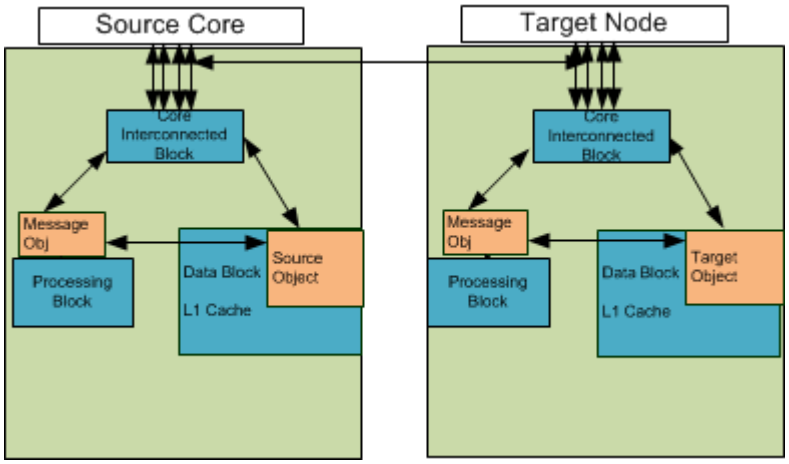

Figure 3. Object Interaction in TBHIN Architecture

Algorithm

I is the source node, K is the destination node and J is the current node
foreach run time **do**
ifJ = I**then**
ifone of the conditions in J is true
then send one datapacket from each of the three area of J
else
send one forward ant from each of the three area of J
end if
end if

foreach message O in J's buffer **do**

if(O→type = ForwardAnt) or (O→type = DataPacket) **then**
NextHop= SelectNextHop(L)
sendL to NextHop
ifNextHop = K**then**
O →type = BackwardAnt
end if

else if O→type = BackwardAnt**then**
findNextHop in J's BackRouting table
sendO to NextHop
IncreasePheromone(NextHop,L)
ifNextHop = I**then**
update averages of packet delays for the correspondingarea
dropO
end if
end if
end for
Execute()
end for

Figure 4. Object Interaction Algorithm

Figure 3 shows processing part of source Core executing instruction message. From the computer's point of view, instructions are the smallest tasks/jobs which cannot be divided further. Operator and operand compose an instruction [11].The message builds up in the source core Interconnected part Buffer waiting to be sent, the message through the central point forward to Interconnected part of the target core, ultimately reaches the target object. After being received at target Core and temporarily stored in buffer queue, cache processes it at appropriate time. This process is mainly based on the object table for the Position/Location of object in message queue. Then the priority of the message will be inserted into the message

queue which will be our future work not focused in this paper. Then message/notice is sent to the target object waiting for processing. At this point process of message sending and receiving has been completed and next (message) processing is carried out by target core . Processing part in reply to pervious message may send the address of the data/ method or invoke that method which is needed by Corel's objI1 to complete the task.

Figure 4 illustrate the object interaction algorithm. In Figure 5 the simulation graph shows the performance of the objects (intra inter triplet interaction). C1T1 represents the first core of triplet1, C2T1 Core 2 of triplet one and C3T3 core 3 of triplet 3. The DDRA algorithm will be used which is already declared for TBHIN.

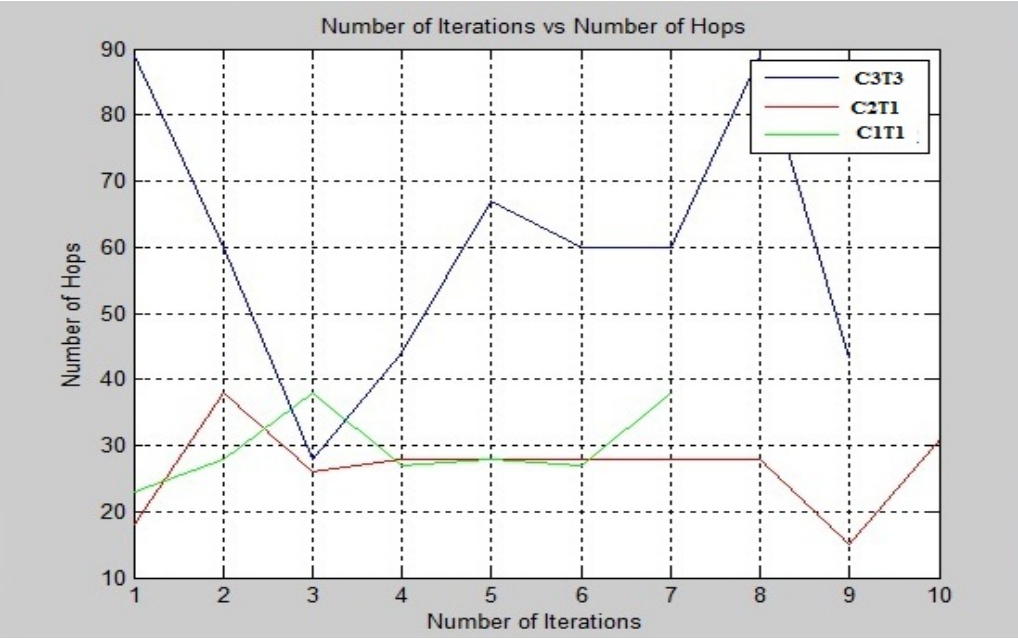

Figure 5. Empirical Graph

5. Related Work

Author Tansel Ersavas in [12] defined mechanism for communicating objects with fuzzy logic technique FOBM, using constraints in invisible message swapping for object manipulation. Authors Takayuki Tachikawa and Makoto Takizawa in [4] describe a protocol based on group communication among objects. Messages delivered between objects in groups are said intra-group communication. Authors P.N.Green and M.D.Edwards in [13] describe reconfigurable hardware objects development model for application used in embedded systems and high level communication between software objects adapting Moose Object-Oriented system. Authors David Unger and Sam S. Admas in [14] expressed channel level messaging in the categories of buffered, raw and streaming for performance measurement of TILE64. Authors MA Liwie and SUN Yihe in [15] articulated that one object can invoke other object's method, after receiving message it also can trigger that method. Author Licheng Xue and Feng Shi in [16] introduced OCCU hardware based scheme for message passing in CMP to communicate between cores in TriBA based on NOC but our work is different from that as it employs purely upper language level objects oriented message mechanism.

6. Conclusion

In this paper we show the interaction between the objects in Triplet-Based Hierarchical Interconnection Network. The result shows that when the object in one triplet needs some methods which are under another core's object guardianship then how such iteration could be placed verses number of hosts. This method could be used for improvement in the performance

and efficiency. In the future work we will work on more levels of TBHIN as the number of cores is increasing rapidly in CMP.

References

[1] Mengxiao L, Weixing J, Xing P, Jiaxin L. *A Parallel Memory System Model for Multi-core Processor*. IEEE International Conference on Networking, Architecture and Storage (NAS). 2009: 219-222.

[2] Mengxiao L, Weixing J, Jiaxin L, Xing P. *Storage Architecture for an On-Chip Multi-Core Processor.* 12th Euromicro conference on Digital System Design, Architecture, Methods and Tools. 2009: 263-270.

[3] I-Hsin C, Che-Rung L, Jiazheng Z, Yeh-Ching C. *Hierarchical Mapping for HPC Application*. Proceedings of the IEEE International Symposium on Parallel and Distributed Processing Workshops and PhD Forum (IPDPSW '11). 2011: 1815-1823.

[4] Takayuki T. and Makoto T. *Object-Based Message Ordering in Group Communication.* Third International Workshop on Object-Oriented Real-Time Dependable Systems. Newport Beach, CA. 1997: 315-322.

[5] Weixing J, Yizhuo W, Zhi H, Junqing Z, Xi L. *Exploring object-level parallelism on chip multi-processors*. Algorithms and Architectures for Parallel Processing Lecture Notes in Computer Science Volume 7440. 2012: 80-89.

[6] Yang W, Guofeng Q. A Multicore Load Balancing Model Based on Java NIO. *TELKOMNIKA Indonesian Journal of Electrical Engineering.* 2012; 10(6): 1490-1495.

[7] Mr Nigel M. Developing a Secure Programming Module to cope with Modern Vulnerabilities. *International Journal of Information & Network Security (IJINS)*. 2012: 1(1): 41-44.

[8] José A. Á, Javier R, José J. F. *From Structured to Object Oriented Programming in Parallel Algorithms For 3D Image Reconstruction*. Proceedings of the 8th workshop on Parallel/High-Performance Object-Oriented Scientific Computing (POOSC '09). 2009: 1-8.

[9] Qiao B, Shi F, and Ji W. *A New Routing Algorithm in Triple-based Hierarchical Interconnection Network.* Proceedings of the First International Conference on Innovative Computing, Information and Control (ICICIC'06). 2006; 1: 725-728.

[10] Yang Z, Feng S, Qi Z, Talpur S,Ziyu L. *Express Router Microarchitecture for Triplet-based Hierarchical Interconnection Network*. IEEE 14th International Conference on High Performance Computing and Communication. 2012: 295-302.

[11] Bin L, Zhi-Chen T, Yu-Jin G. *Triplet-based architecture and its process migration mechanism.* International Conference on Machine learning and Cybemetics. 2009; 5: 2754-2758

[12] Tansel E. *Enhancing Object Communication Mechanisms.* Technology of Object-Oriented Languages and Systems (TOOLS 27). 1998: 232.

[13] P.N.Green and M.D.Edwards. Object Oriented development method for reconfigurable embedded Systems. *IEE Proc-Comput. Digit. Tech*. 2000; 147(3): 153 - 158

[14] David U, Sam S. A. *Hosting an Object Heap on manycore Hardware: An Exploration.* DLS '09 the 5th symposium on Dynamic languages. Orlando, FL, USA. 2009: 99-11.

[15] MA L, SUN Y. Object-Oriented System-on-Network-on-Chip Template and Implementation: H.263 Case Study. *Tsinghua Science and Technology.* 2008; 13(1): 98-105.

[16] Licheng X, Feng S. *Low overhead object communication scheme in CMP implementation of object-oriented programming*. Proceedings of the IEEE 15th International Conference on Parallel and Distributed Systems (ICPADS). 2009: 925-930.