ISSN:0975-9646

Poonam Karande et al, / (IJCSIT) International Journal of Computer Science and Information Technologies, Vol. 5 (1) , 2014, 636-639

Task Management for Heterogeneous Multi-core Scheduling

Poonam Karande, S.S.Dhotre, Suhas Patil

Computer Engineering Department BVDUCOE
Pune-43(India)

***Abstract*- Multi-core task scheduling is a challenging problem with precedence constraint & non-preemptive task. The existing system uses heterogeneous dual-core scheduling as well as heterogeneous multi-core scheduling. In this paper we propose a new approach towards the heterogeneous multi-core scheduling with dispatcher schemas, where heterogeneous co-processor can be multiple means that one processor & multiple heterogeneous co-processors. The Processor is responsible for handling control signal & heterogeneous co-processor responsible for data computation. Because of multiple co-processors dispatcher is needed to determine which subtask dispatch to which co-processor. We will also implement migration policies. With different migration policies it will increase the use of resources and also improve efficiency and performance.**

***Keywords*- Task scheduling, Multi-core, Heterogeneous.**

I. Introduction

Most Embedded systems which make a system more dedicated for an application or part of large system are provided with a heterogeneous multi-core system, where one general purpose processor and more than one synergistic co-processor [1], [2], [3] to increase performance and power capability. To fulfil the user requirement, the quality of service for applications has to be met. Different applications are functioning on such a system with dynamic workloads such as communication (networking) and personal devices (video), as it varies the number of simultaneously executing tasks. Thus, an interactive and predictable online scheduling algorithm is an essential system specification for a heterogeneous multi-core system. Still the scheduling problem of heterogeneous multi-core systems is complex because of precedence constraints of tasks and non-preemptive execution of tasks in the synergistic co-processor.

In the past ten years, number of researches has continued synchronization protocols to determine the task scheduling problem in multi-core systems by handling the non-preemptive co-processor as a resource. In this the handling of priority inversion bring on to low system utilization [4], [5].The low system utilization is because of task execution with lower priority on the processor prohibited even when the processor is unproductive, if a task with higher priority is executed in the co-processor. To make better system utilization, authors explained frameworks for extending previous protocols to such system and composed the prohibited constraints for signal processing co-processors[6],[7].

Earlier research has enhanced multiprocessor scheduling algorithm for homogeneous multi-core system [8], [9], [10], [11] to prevent the priority inversion management problem. Still these algorithms are not suitable for heterogeneous multi-core system. The explanation behind this is that the both processor and co-processor are asymmetric and the co-processor is not appropriate for pre-emptive task execution because of significant pre-emption overhead. This overhead occurs due to number of registers, pipeline stages [12], [13], and cache flushes [14].

This effort is stimulated by need for heterogeneous multi-core system for an online scheduler, and also hazard enforced by the accommodation between handling of priority inversion and enhancement of system utilization. In this proposed system the novel idea is a heterogeneous multi-core scheduling algorithm with a key technique dispatcher mechanism having different schemas. In heterogeneous multi-core, one general purpose processor and more than one synergistic co-processor which execute the different tasks with high speeds. In this, general purpose processor is responsible for handling control signals and synergistic co-processors responsible for data computation. A dispatcher mechanism is used because of multiple co-processors dispatcher is needed to determine which ready subtask dispatch to which co-processor. In that dispatcher mechanism can be implemented using different dispatcher schemas such as global schema, partition schema, and hybrid schema [15]. This paper also contributes towards task migration policies [24] which will improve the performance.

This paper contributes three aspects:

- It gives better or high performance than the previous scheduling.
- It uses dispatcher mechanism to dispatch different subtask.
- It uses task migration policies to increase the utilization of resources.

The rest of this paper is organized as follows: In section 3 we will analyse the previous heterogeneous dual-core architecture as well as various multi-core scheduling approaches. Section 4 proposes new heterogeneous multi-core system for scheduling. Section 5 proposes algorithm for heterogeneous multi-core scheduling. We draw a conclusion in section 6.

II. Background And Motivation

Fast and efficient performance is the necessity of today’s technology. Hard real time system has the critical factor of time i.e. hard real-time system should required task to be complete within accurate time. Applications such as satellite launching require performing thousands of tasks with much concern about time. An embedded system also requires multitasking within integrated circuit. Today's applications based on either soft real-time system or hard real-time system involving multiplicity of tasks within time constraint. So we have to migrate from dual-core to multi-core system along with heterogeneity of co-processors which improves the performance of the system.

III. EXISTING SYSTEM

Bj¨orn Andersson et al. has proposed the technique for Assigning Real-Time Tasks on Heterogeneous Multiprocessors with Two Unrelated Types of Processors [16]. In this paper they present a new algorithm is FF-3C, which offers low time-complexity as well as good performance. They also present number of extensions to FF-3C; all extensions offer the same Time-complexity and better performance as that of FF-3C. And they also offer improved average-case performance.

I-hong hou and p. r. kumar has proposed Scheduling Periodic Real-Time Tasks with Heterogeneous Reward Requirements [17]. In this paper they have explained the problem of scheduling periodic real-time tasks which has precise minimal reward requirements. They also consider positions where tasks generate instance that can be provided subjective service times before their deadlines, and get rewards based on the service times received by the instance of the task. They also wind up that this model is appropriate with the estimated computation models also increasing reward with expanding service models. They also add to accomplish different reward requirements of Different tasks. This provides better equity as also as allowing fine-grained accommodation between tasks.

Hsiang-Kuo Tang et al. have proposed the approach for Combining Hard Periodic and Soft Aperiodic Real-Time Task Scheduling on Heterogeneous Compute Resources [18]. In this paper they spotlight on scheduling soft aperiodic tasks along with hard time limit periodic Tasks with constraints on heterogeneous real-time systems. Even they also explain a method to upgrade aperiodic task responsiveness without partitioning periodic task time limit guarantees, by first scheduling periodic tasks offline, then dynamically scheduling aperiodic tasks in the remaining resource slack time.

Zhe Wang et al. propose Temperature-aware Task Partitioning for Real-Time Scheduling in Embedded Systems[19].In this paper they have explained task partitioning as an impressive manner to reduce the crest temperature in embedded systems running either a set of periodic heterogeneous tasks with identical period or periodic heterogeneous tasks with individual period.

Pengliu Tan et al. present A Hybrid Real-Time Scheduling Approach on Multi-Core Architectures [20]. In this paper for real-time task on homogeneous multi-core architecture they provide a hybrid scheduling approach. This approach uses the top level and a bottom level scheduling scheme. In the top level scheme, for each scheduling policy a sporadic server is committed. To schedule the dispatched tasks according to its scheduling policy each sporadic server is used. In the bottom level scheme, a RM OS scheduler is adopted to manage and schedule the top level sporadic servers.

Li Wenjing and Wang Lisheng have proposed Energy Considered Scheduling Algorithm Based on Heterogeneous Multi-core Processor [21]. In this paper they don't only consider the tasks execution time but also energy consumption in embedded system and they also explain three scheduling principles such as the minimum earliest finish time, load balancing and low energy consumption which are convenient to independent task scheduling in heterogeneous multi-core processor. And also present new algorithm ECSA which consider these three principles.

Derong Liu et al. have proposed Pipeline-based Scheduling for Heterogeneous Multi-core Systems [22]. In this they have proposed a pipeline-based scheduling algorithm to progress the throughput for heterogeneous multi-core. This algorithm reduces the communication and idle time by assigning most parent-child task pairs on the same processor, and equity processors work time through a balance condition.

IV. PROPOSED SYSTEM

In this section, we have proposed a heterogeneous multi-core scheduling algorithm for task with a key technique called a dispatcher mechanism.

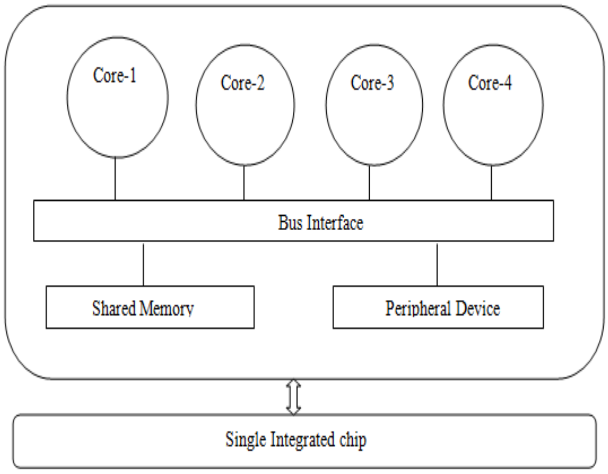

Figure 1: Overview of Multi-core system

Firstly, heterogeneous multi-core system is an integrated circuit(IC) to which two or more heterogeneous processors have been attached to decrease power consumption, enhanced performance and more efficient processing of multiple tasks.

Here processor is a component that can read and executes program instructions; those program instructions tell the processor what to do such as read data, send data. A processor component is a central processing unit (CPU), in which two or more heterogeneous processors are attached to an integrated circuit. And the cores or processors share common memory, common bus interface as well as peripherals. Heterogeneous multi-core is used across many application domains such as embedded system, network, digital signal processing and graphics etc. The performance improvement is gained by the use of heterogeneous multi-core system is much depends on the software algorithm which are used and their implementation.

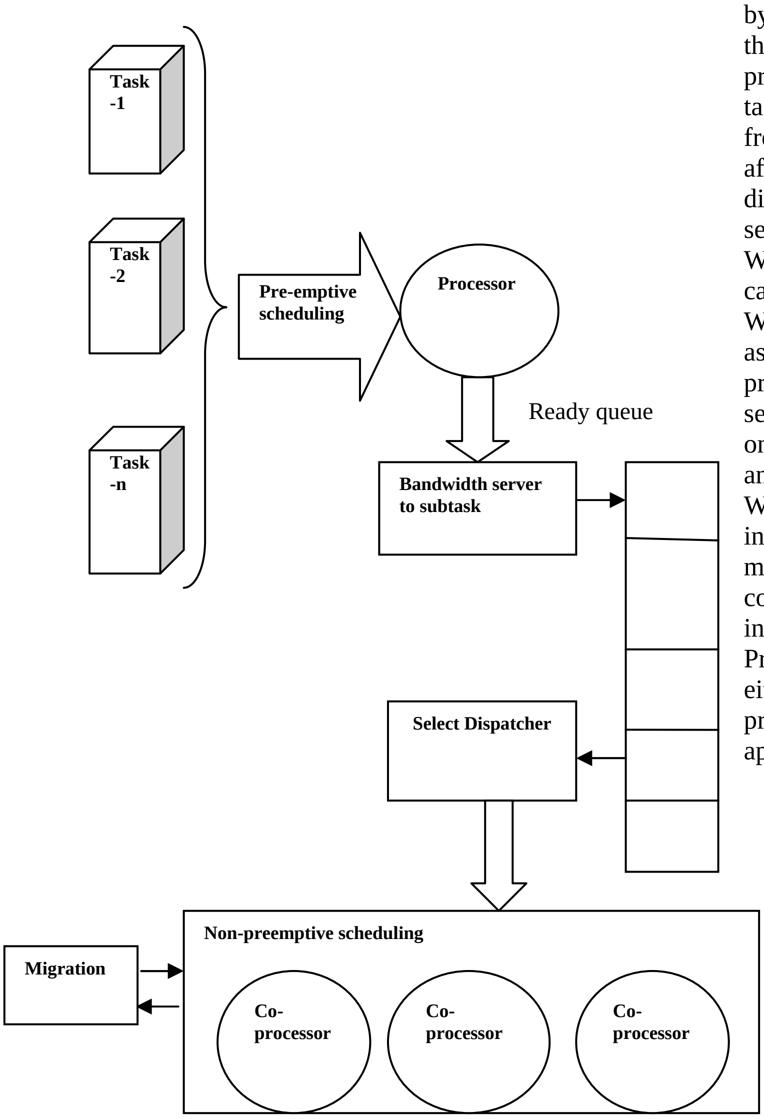

Figure 2: System workflow

The proposed system is a heterogeneous multi-core scheduling algorithm which applying dispatcher mechanism with different dispatcher schemas [15]. In heterogeneous multi-core system Processor is responsible for handling control signals and synergistic co-processors responsible for data computation. Task execution which is done in a processor is preemptive here T1, T2.......Tn are numbers of input tasks given to the processor and task execution in co-processor is usually non-preemptive, a task might have large blocking time to schedule. The solution for this problem, we enter preemption point into co-processor subtask so that the co-processor can be 'semipreemptive' for accommodating a bandwidth server and bounding blocking times of subtasks. When there are multiple synergistic co-processors a dispatcher is needed to determine which subtask dispatch to which co-processor. Dispatcher mechanism can be implemented using global schema, partition schema and hybrid schema.

In Global dispatcher schema, each co-processor subtask in the proposed system is maintained by an appropriate sever. When the job of the subtask is ready then the corresponding server sets the completion deadline into the job & indexed by total order includes it into the global ready queue then the highest priority tasks are given to heterogeneous co-processors to process. No processor is ever idled when a task is ready to execute. The task executing might jump from one processor to another as a processor might get idle after executing a task. Migration is allowed in the global dispatcher schema. The specific aspect of bandwidth servers implementation under multi-core present in [23]. With bandwidth server, Job of each co-processor subtask is captured as a sporadic job.

With Partition dispatcher schema, each server is first assigned to an appropriate coprocessor then all the co-processors subtasks of a task are then executed by a specific server in the corresponding heterogeneous co-processor only. Here the task cannot switch between the co-processor and must complete executing on the same co-processor.

With Hybrid dispatcher schema, most servers are divided into one individual coprocessor and remaining servers might migrate between heterogeneous coprocessors and the corresponding coprocessor subtasks might be split to increase performance.

Proposed system also implement the task migration which either involves migrating the waiting tasks from busy co-processor to the newly idle co-processor or migrating aperiodic tasks to idle co-processor.

V. Heterogeneous Multi-Core Scheduling Algorithm [15]

- Create the sample task which will run in the background and various options like global, partition, hybrid to select dispatcher technique. Code to get those tasks for scheduling and Different threads generation for core.
- Order the tasks to maintain precedence constraints. Bandwidth server model, where server size is calculated according to heterogeneous co-processors.
- Schedule the task to the processor and also create the dispatcher so that bandwidth server is assigned to co-processor.
- Schedule the task and subtasks between different cores with the help of dispatchers so that processors are scheduled by maintaining precedence constraints and non-preemptive task executions in the co-processors.
- Global partition require that different sub-task when executing will migrate co-processor, so we will implement a migration policies which maintains the system schedulable in spite of migration overhead generated.

VI. Conclusion

This paper has explored the previous heterogeneous dual-core scheduling as well as multi-core scheduling. To extend this scheduling, we have proposed a new scheduling algorithm called heterogeneous multi-core scheduling algorithm for task with a key technique is called as a dispatcher mechanism with different schemas such as global schema, Partition schema, and hybrid schema with consideration of precedence constraints as well as non-preemptive execution on the co-processor. This paper has also implemented Migration policies. With migration policies it will reduce response times for tasks and also increased complete system utilization with a guarantee of real-time deadlines. This result also can efficiently applicable for future mobile systems.

REFERENCES

[1] Texas Instruments, Inc., "OMAP3 Platform," technical report,Texas Instruments, http://www.ti.com/lit/ml/swpt024b/ swpt024b.pdf, 2009.

[2] Texas Instruments, Inc., "OMAP4 Platform," technical report,TexasInstruments,http://www.ti.com/lit/ml/swpt034b/swpt034b.pdf, 2011.

[3] Qualcomm, Inc., "Snapdragon," technical report, Qualcomm,http://www.qualcomm.com/media/documents/snapdragons4-processors-system-chip-solutions-new-mobile-age, 2011.

[4] L. Sha, R. Rajkumar, and J. Lehoczky, "Priority Inheritance Protocols: An Approach to Real-Time Synchronization," *IEEE Trans. Computers, vol. 39, no. 9, pp. 1175-1185, Sept. 1990.*

[5] T.P. Baker, "Stack-Based Resource Allocation Policy for Real-Time Process," *Proc. Real Time Systems Symp.,* 1990.

[6] P. Gai, L. Abeni, and G. Buttazzo, "Multiprocessor dsp Scheduling in System-on-a-Chip Architecture," *Proc. Euromicro Conf. Real- Time Systems, 2002.*

[7] K. Kim, D. Kim, and C. Park, "Real-Time Scheduling in Heterogeneous Dual-Core Architecture," *Proc. Conf. Parallel and Distributed Systems*, 2006.

[8] L. Benini, D. Bertozzi, A. Guerri, and M. Milano, "Allocation, Scheduling and Voltage Scaling on Energy Aware MPSoCs," *Proc. Conf. Integration of AI and OR Techniques in Constraint Programming for Combinatorial Optimization Problems*, 2006.

[9] M. Kim, S. Banerjee, N. Dutt, and N. Venkatasubramanian,"Design Space Exploration of Real-Time Multi-Media mpsocs with Heterogeneous Scheduling Policies," *Proc. Conf. Hardware/Software Codesign and System Synthesis*, 2006.

[10] C.-F. Kuo and Y.-C. Hai, "Real Time Task Scheduling on Heterogeneous Two-Processor Systems," *Proc. Conf. Algorithms and Architectures for Parallel Processing,* 2010.

[11] B. Andersson, G. Raravi, and K. Bletsas, "Assigning Real-Time Tasks on Heterogeneous Multiprocessors with Two Unrelated Types of Processors," *Proc. Conf. Real-Time Systems Symp*., 2010.

[12] Texas Instruments, Inc., "DSP/BIOS II Timing Benchmarks on the TMS320C54x DSP," technical report, Texas Instruments, http:// focus.ti.com, 2000.

[13] K.-Y. Hsieh, Y.-C. Lin, C.-C. Huang, and J.-K. Lee, "Enhancing Microkernel Performance on VLIM DSP Processors via Multiset Context Switch," *J. Signal Processing Systems*, vol. 51, no. 3, pp. 257- 268, 2008.

[14] F.M. David, J.C. Carlyle, and R.H. Campbell, "Context Switch Overheads for Linux on Arm Platforms," *Proc. Workshop Experimental Computer Science*, 2007.

[15]Ya-Shu Chen, Member, IEEE, Han Chiang Liao, and Ting-Hao Tsai, Student Member, IEEE,"Online Real-Time Task Scheduling in Heterogeneous Multi-core System-on-a-Chip" *IEEE Trans on parallel and distributed systems*, vol. 24, no. 1, January 2013.

[16] Bj¨orn Andersson, Gurulingesh Raravi and Konstantinos Bletsas, "Assigning Real-Time Tasks on Heterogeneous Multiprocessors with Two Unrelated Types of Processors,"Proc. *Conf. Real-Time System Symp.*, 2010.

[17] I-hong hou, p. r. kumar,"Scheduling Periodic Real-Time Tasks with Heterogeneous Reward Requirements," 2011 32nd *IEEE Real-Time Systems Symposium.*

[18] Hsiang-Kuo Tang, Parmesh Ramanathan, Katherine Compton," Combining Hard Periodic and Soft Aperiodic Real-Time Task Scheduling on Heterogeneous Compute Resources," 2011 *IEEE International Conference on Parallel Processing.*

[19] Zhe Wang, Sanjay Ranka and Prabhat Mishra," Temperature-aware Task Partitioning for Real-Time Scheduling in Embedded Systems, " 2012 *IEEE 25th International Conference on VLSI Design.*

[20] Pengliu Tan, Jian Shu and Zhenhua Wu," A Hybrid Real-Time Scheduling Approach on Multi-Core Architectures," *journal of software, vol. 5, no. 9, september* 2010.

[21] Li Wenjing Wang Lisheng," Energy-Considered Scheduling Algorithm Based on Heterogeneous Multi-core Processor," 2011 *International Conference on Mechatronic Science, Electric Engineering and Computer August* 19-22, 2011, Jilin, China.

[22] Derong Liu, Ming'e Jing, Yuwen Wang, Zhiyi Yu, Xiaoyang Zeng, Dian Zhou," Pipeline-based Scheduling for Heterogeneous Multi-core Systems,"*IEEE* 2012.

[23] S. Baruah and G. Lipari, "Executing Aperiodic Jobs in a Multiprocessor Constant-Bandwidth Server Implementation,"*Proc.Euromicro Conf. Real-Time Systems,* 2004.

[24] Kedar M. Katre, Harini Ramaprasad, Abhik Sarkar, Frank Mueller," Policies for Migration of Real-Time Tasks in Embedded Multi-Core Systems".