An agent based Distributed thermal balancing –Task migration

[1]T L Rekha Reddy, [2]Mr C K Raju, [3]Dr. M Z Kurian

[1] 4th sem, M.Tech (VLSI & Embedded Systems), SSIT, Tumkur, Karnataka

[2] Asst. Prof, Dept of CS&E, SSIT, Tumkur, Karnataka

3 HOD, Dept of E&C, SSIT, Tumkur, Karnataka

Abstract**—** The system reliability, performance, cost, and leakage power in deep sub micrometer era have a significant impact by thermal hotspots and temperature gradients. As the system complexity increases, it is very difficult to perform thermal management in a centralized approach due to state explosion and the overhead of monitoring the entire chip. In this work, a framework for distributed thermal management in many-core systems where balanced thermal profile can be achieved by proactive task migration among neighbouring cores is proposed. The framework has a low cost agent residing in each core that observes the temperature and communicates with its nearest neighbour for task migration and exchange. Unnecessary migration are avoided by choosing only those migration requests that will not result in generating thermal emergency.

Key Words**—** Distributed control, dynamic thermal balancing, multi-agent, task migration.

I. Introduction

WITH the increased number of transistors integrated on a single chip, the current multi-core technology may soon progress to hundreds or thousands of cores era. While the ever-increasing complexity and computing capability of multicore or many-core technology delivers extraordinary performance, they have to face the significant power and thermal challenges. High power and temperature densities, process imperfections and reduced voltage margins have made the systems much more vulnerable to both permanent and transient faults. Elevation in temperatures (i.e. thermal hot spots) and temperature gradients bring new challenges in reliability, leakage power ,performance, cooling costs and timing. Cost of cooling increase at a super-linear rate, which requires designing for temperature margins that are lower than the worst-case. Hot spots accelerate the failure mechanisms such as stress migration, electromigration and dielectric breakdown, which cause permanent device failures. So there is a need remove the heat which is an undesirable by-product of the faster electronic devices and systems used in daily life. The goal of heat removal is to keep the operating temperature within an acceptable range where the multicore systems can function as expected and reliably.

For multi-core systems many dynamic thermal management techniques such as stop and go ,clock gating, DVFS, thread migration have been proposed . All these techniques ensure the system running under a fixed safe temperature constraint. The presented techniques are centralized approaches. They require a controller that monitors the temperature and workload distribution of each core on the entire chip and global decisions of resource allocation are taken. First of all, as the number of cores increases resource allocation and management problem also increases due to complexity. Second, controller communication has a significant impact on the system performance and energy. As a single controller monitors and issues the required commands to each core communication overhead increases. Such overhead will affect the speed of data communication among user programs and also consume more power. Lastly as the size and the complexity of the many-core system increase the communication latency between the central controller and the cores increases, this leads to a delayed response and sub-optimal control.

This work, propose a framework of an agent based distributed thermal management where balanced thermal profile can be achieved by thermal-aware task migrations among neighboring cores.

2*2 mesh connected core architecture is targeted in this work. It will also be referred to as core or PE in this work. All the processors and routers are connected by an on-chip network where information is communicated via packet transmission. Each processing element (PE) consists of a low cost agent. The agent observes the temperature of the PE through negotiation and communication it exchanges the tasks with its nearest neighbors . The goal of the proposed task migration is to match the PE's heat removal capability to its workload (i.e., the average power consumption) and at the same time create a good mix of high power (i.e., "hot") tasks and low power (i.e., "cool") tasks running on it. The proposed technique is referred as distributed thermal balancing migration (DTB-M).

II. Related Work

1. *"Techniques for multicore thermal management.*

This paper explores different thermal management techniques
that exploit the distributed nature of multicore processors. These techniques are classified in terms

of core throttling policy and process migration policies.

The DVFS policy is also proposed here and it involves more of a continuous adaptive scheme. By enabling a continuous range of frequency and voltage combinations to reduce power consumption. Thus DVFS policy is not as simple as the stop-go mechanism, a set point slightly below the thermal threshold is used and use a PI controller to adaptively control the frequency and voltage levels to aim towards the target threshold. DVFS mechanism has a higher design cost than the rudimentary stop-go mechanism due to the complexity of implementing a flexible phase-lock loop (PLL) and voltage scaling capabilities.

2. *"Temperature aware task scheduling in MPSoCs"*

This work, explores the benefits of thermally aware task scheduling for multiprocessor systems-on-a-chip(MPSoC).This also designs and evaluates OS-level dynamic scheduling policies with negligible performance overhead. Compared to state -of- art schedulers better temporal and spatial thermal profiles can be achieved using simple to implement policies that make decisions based on temperature measurements. Reactive strategies such as dynamic thread migration with scheduling policies are improved. This way, hot spots and temperature variations are minimised, and the performance cost is significantly reduced.

Dynamic Thread Migration is an MPSoC thermal management method that migrates threads from hot processors to cooler ones. For minimizing the performance impact of thread migration, Heat-and-Run is proposed for loading the cores as much as possible and migrating workload when critical temperature values are observed

3. *"Utilizing predictors for efficient thermal management in multiprocessor SoCs"*

Conventional thermal management techniques are reactive in nature; that is, they take action only after temperature reaching a predetermined threshold value. Such methods do not always minimize and balance the temperature on the chip. This paper says how to use predictors for forecasting future temperature and workload dynamics, and proposes proactive thermal management techniques for multiprocessor system-on-chips (MPSoCs). The predictors are autoregressive moving average (ARMA) modelling and look-up table based predictors.

4. *"Predictive dynamic thermal management for multicore systems"*

In this paper, a Predictive Dynamic Thermal Management (PDTM) based on Application-based Thermal Model (ABTM) and Core-based Thermal Model (CBTM) in the multicore systems is proposed . ABTM predicts future temperature based on the application specific thermal behaviour, while CBTM estimates core temperature pattern by steady state temperature and workload. It requires runtime adaptation.

4. *"Thermal-aware job allocation and scheduling for three dimensional chip multiprocessor"*

This paper, proposes a thermal management algorithm for three-dimensional (3D) chip multiprocessor (CMP). The proposed thermal-aware job allocation and scheduling algorithm assigns hot jobs to the cores near to the heat sink and cool jobs to the cores far from the heat sink. The effect of the proposed algorithm on a 3D-CMP system is that, the heat from hot jobs is removed from the chip faster than temperature-aware methods. Therefore it is possible to keep the chip cooler and in better thermal condition.

5. *"Route packets, not wires: On-chip interconnection networks"*

A tile-based network-on-chip (NoC) architecture is targeted here. Each tile is a processor with dedicated memory and an embedded router. The concept of routing packets and not using wires has been extracted from this paper which includes on-chip interconnection networks in place of ad-hoc global wiring structures the top level wires on a chip and facilitates modular design. With this approach, system modules (processors, memories, peripherals, etc...) communicate by sending packets to one another over the network.

III. System Design

A tile-based 2*2 processor connected in mesh topology is targeted here. It will also be referred to as core or PE in this work. The cores that can reach to each other via one-hop communication are referred as the nearest neighbors. The proposed DTB-M algorithm moves tasks among nearest neighbors in order to reduce overhead and minimize the impact on the communication bandwidth.

The proposed DTB-M algorithm is implemented in each core which performs thermal-aware task migration.

DTB-M policy:

The main idea of the DTB-M policy is to exchange tasks between neighbouring PEs, so that each PE can get a set of tasks that produces fewer hotspots.

The DTB-M policy basically can be divided into 3 phases: temperature checking, information exchange and task migration. Fig.1 shows the flowchart of the DTB-M execution in the ith core[1]. A DTB-M agent could be in either master mode or slave mode. A task migration request is initiated by DTB-M while the DTB-M slave responds to a task migration request. A DTB-M agent is in slave mode by default. It will enter the master mode only if the below condition is true.

The local temperature T_i reaches a threshold T_m . In this case, hotspots are generated, and the DTB-M agent will first throttle the processor to let it cool down before continue to execute. This is done by migrating the task to the neighbouring cores of low temperature

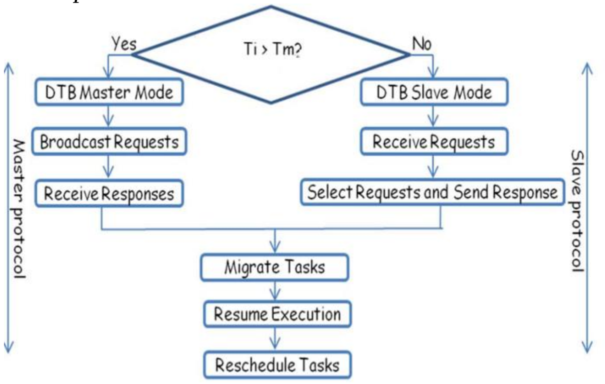

Fig 1: Master-slave execution protocol

The DTB-M master sends task migration requests to its nearest neighbours which are of low temperature.

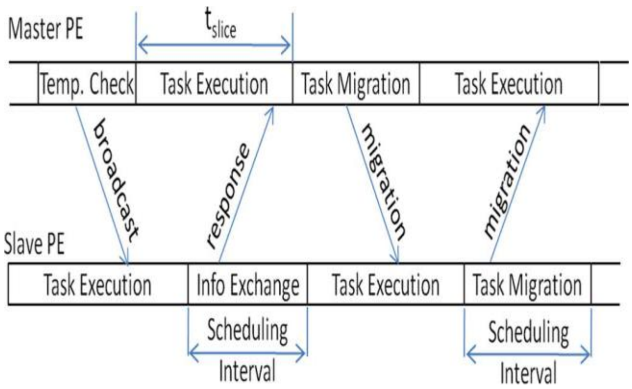

Fig 2: Master- slave communication

The asynchronous communication between master and slave agents will take place as explained in Fig.2 [1]. It shows a complete execution cycle of DTB-M policy starting from condition check phase to task migration. When an agent first enters its scheduling interval and becomes a master, it broadcasts a migration request in its neighbourhood. The slave will respond to the request by accepting it. If there is no request, the PE resumes normal execution in next time slice. In case of multiple master requests, the slave selects a master which has the highest average power consumption. The slave is then locked to this master until it is released by the master. After receiving the response, the master decides which tasks to migrate during its next scheduling interval and sends the migration command to slave. The tasks are migrated from master to slave at this time. After sending a response, the slave ignores any possible incoming requests from other master agents until it receives the migration command from the original master. Tasks can be migrated from slave to master at this time, which marks the end of DTB-M policy cycle.

Distributed task migration policy

TPM policy:

TPM policy present at slave DTB-M agent determines whether the local peak temperature will exceed the thermal threshold after exchanged.

TPM policy at master DTB-M agent selects the offer that enables it to move out the task with highest power consumption.

IV. FUTURE WORK

A neural network prediction model can be used to predict the future temperature. Task migration can be done based on the predicted future temperature and also on work load

V. CONCLUSION

This project, proposes a distributed thermal management framework for many-core systems. In this framework, no centralized controller is needed. Each core has an agent which monitors the core temperature, communicates and negotiates with neighboring agents to migrate and distribute tasks evenly across the system. The agents use DTB-M policy for task migration. Two types of results are expected from this proposed project first one is when the core temperature goes above the threshold temperature the core communicates with the neighbouring core only if the neighbor core temperature is less than the threshold task will migrate. The second is when all the cores temperature goes above the threshold then the system will stop executing for the designed period .i.e. until the system temperature goes below the threshold.

ACKNOWLEDGMENT

I am deeply indebted and I would like to express my sincere thanks to our beloved Principal **Dr. K.A.Krishnamurthy,** for providing me an opportunity to do this Project.

My special gratitude to **Dr. M.Z.Kurian,** HOD, department of E & C, S.S.I.T for his guidance, constant encouragement and wholehearted support.

My Sincere thanks to my guide **Mr C K Raju**, Asst Prof, Department of CS&E, for his guidance, constant encouragement and wholehearted support.

Finally, yet importantly, I thank almighty for everything and would like to express my heart-full thanks to my beloved parents for their blessings, my friends and classmates for their help.

REFERENCES

[1] Yang Ge, "A Multi-Agent Framework for Thermal Aware Task Migration in Many-Core Systems" *IEEE transactions on very large scale integration (vlsi) systems, vol. 20, no. 10*, october 2012

[2] J.Donald and M. Martonosi, "Techniques for multicore thermal management: Classification and new exploration," in *Proc. Int. Symp. Comput. Arch. (ISCA)*, 2006, pp. 78–88.

[3] A.Coskun, T. Rosing, and K. Whisnant, "Temperature aware task scheduling in MPSoCs," in *Proc. Design Autom. Test Euro. (DATE)*, 2007, pp. 1659–1664.

[4] A.Coskun, T. Rosing, and K. Gross, "Utilizing predictors for efficient thermal management in multiprocessor SoCs," *IEEE Trans. Comput.- Aided Design Integr. Circuits Syst.*, vol. 28, no. 10, pp. 1503–1516, Oct. 2009.

[5] I.Yeo, C. Liu, and E. Kim, "Predictive dynamic thermal management for multicore systems," in *Proc. Design Autom. Conf. (DAC)*, 2008, pp. 734–739

[6] S.Liu, J.zhang, Q. Wu, and Q. Qiu, "Thermal-aware job allocation and scheduling for three dimensional chip multiprocessor," in *Proc. Int. Symp. Quality Electron. Design (ISQED)*, 2010, pp. 390–398.

[7] W.Dally and B. Towles, "Route packets, not wires: On-chip interconnection networks," in *Proc. Design Autom. Conf.*, 2001, pp. 684–689.