Dual Monitoring Communication for Self-Aware Network-on-Chip: Architecture and Case Study

Liang Guang, University of Turku, Finland
Ethiopia Nigussie, University of Turku, Finland
Juha Plosila, University of Turku, Finland
Hannu Tenhunen, Royal Institute of Technology, Sweden

ABSTRACT

Self-aware and adaptive Network-on-Chip (NoC) with dual monitoring networks is presented. Proper monitoring interface is an essential prerequisite to adaptive system reconfiguration in parallel on-chip computing. This work proposes a DMC (dual monitoring communication) architecture to support self-awareness on the NoC platform. One type of monitoring communication is integrated with data channel, in order to trace the run-time profile of data communication in high-speed on-chip networking. The other type is separate from the data communication, and is needed to report the run-time profile to the supervising monitor. Direct latency monitoring on mesochronous NoC is presented as a case study and is directly traced in the integrated communication with a novel latency monitoring table in each router. The latency information is reported by the separate monitoring communication to the supervising monitor, which reconfigures the system to adjust the latency, for instance by dynamic voltage and frequency scaling. With quantitative evaluation using synthetic traces and real applications, the effectiveness and efficiency of direct latency monitoring with DMC architecture is demonstrated. The area overhead of DMC architecture is estimated to be small in 65nm CMOS technology.

Keywords: Dual Monitoring Communication (DMC), Monitoring Communication, Network-on-Chip (NoC), Run-Time Management, Self-Awareness

INTRODUCTION

With the sizes of transistors constantly decreasing, modern chips are able to integrate a large number of processing cores. For instance, Vangal et al. (2008) presented an 80-core processor, and Truong et al. (2009) presented a 167-processor computing platform. In the near future, on-chip parallel processing will be running on 1000-core chips (Asanovic et al., 2006). Clearly, how to utilize this large amount of processing power becomes an important topic for integrated circuit and communication system design.

The dynamics of massively parallel on-chip computation platforms pose major challenges to the development of such systems.
For one thing, parallel on-chip systems may need to run different applications, with various performance requirements. The details of applications are likely to be unknown at the design time. Secondly, a diversity of software might run on the system, which is likely to have different resource management schemes. Last but not least, the status of hardware becomes anything but predictable with the small transistor size. Variations (Miranda et al., 2009), for instance, significantly influence the hardware performance. Variations refer to the deviation of transistor sizes or metal wire dimensions from the nominal values. With considerable hardware variations, the system performance, such as speed and power consumption, and system dependability become dynamic and unpredictable.

Autonomic computing (Horn, 2001) is a promising paradigm to cope with the dynamics of massively parallel processing. Self-adaptive computing belongs to the broad spectrum of autonomic computing, but has a narrower conceptual scope. Paraphrased from Salehie and Tahvildari (2009), a self-adaptive system evaluates its own behavior and changes the behavior when the evaluation indicates failure to achieve functional goals or performance optimization is achievable.

This article addresses a specific aspect of self-adaptive computing: self-awareness. As the name itself suggests, a self-aware system is aware of its internal state (Hinchey & Sterritt, 2006). Self-awareness, as an essential part of self-adaptive computing, ensures the visibility of system status, based on which adaptation and optimization can be performed. To provide self-awareness, monitoring and diagnostic operations shall be designed in the system, which can dynamically trace the performance parameters and failures. In embedded system domain, common operations include the monitoring of execution speed, power and energy, errors and failures, as well as temperature.

The work proposes a DMC (dual monitoring communication) architecture to support self-awareness in Network-on-Chips (NoCs). NoC is a scalable communication architecture for massively parallel, high-performance on-chip processing (Jantsch & Tenhunen, 2003; Vangal et al., 2008). Components are modularized as nodes in a network, which supports communication between parallel processing units. DMC provides the communication infrastructure for diagnostic and monitoring operations on NoCs. One monitoring communication is integrated with data channel, to trace the run-time profile, e.g., data value and latency, of data communication. The other monitoring communication is separate from the data channel, and is used to report monitoring information and transmit reconfiguration commands if necessary. DMC architecture is not dependent on network topologies and sizes, flow control schemes or other adaptive techniques, thus it is scalable and generically applicable to different NoC architectures.

A case study, direct latency monitoring on mesochronous Network-on-Chip (NoC), is presented to demonstrate DMC architecture. Mesochronous timing is a special form of globally asynchronous locally synchronous (GALS) timing (Dally & Poulton, 1997), in which different components run with the same frequency but unknown phase differences. The utilization of GALS timing for the NoC design provides significant improvement in design efficiency and power efficiency compared to synchronous design (Amde, Felicijan, Efthymiou, Edwards, & Lavagno, 2005), by eliminating the need of a global clock. Mesochronous NoC architecture is widely used in existing NoC prototypes, for instance the 80-tile Intel TeraFLOPS processor (Vangal et al., 2008). For such systems, accurate run-time latency monitoring is challenging, since there is no global system clock. In this case study, the message latency is directly traced by the integrated monitoring communication, while the separate monitoring communication reports the latency to the supervising monitor. Local congestion is also monitored. The latency measurement is performed by distributed hardware, which is small and fast. It is a novel technique to monitor network performance in a scalable manner, and can be utilized for effective power/performance tradeoff. Quantitative
Semantic Interoperability for Enhancing Sharing and Learning through E-Government Knowledge-Intensive Portal Services
[www.igi-global.com/article/semantic-interoperability-enhancing-sharing-learning/44676?camid=4v1a](www.igi-global.com/article/semantic-interoperability-enhancing-sharing-learning/44676?camid=4v1a)