

Distributed Sensor Data Processing for Many-cores

Jia Zhao, Russell Tessier and Wayne Burleson

Department of Electrical and Computer Engineering

University of Massachusetts, Amherst MA, USA

{jiazhao, tessier, burleson}@ecs.umass.edu

ABSTRACT

Future many-core systems will rely heavily on a wide variety of sensors which provide run-time information about on-chip environment and workload. In this paper, a new dedicated infrastructure for distributed sensor processing for many-core systems is described. This infrastructure includes a sparse array of dedicated processors which evaluate on-chip sensor data and a two-level hierarchical network-on-chip (NoC) which allows for efficient sensor data collection. This design is evaluated using benchmark driven simulations for a three-dimensional (3D) stack, necessitating inter-layer sensor data communication. The experimental results for up to 1024 cores indicate that for typical sensor data collection rates, one sensor data processor (SDP) per 64 cores is optimal for sensor data latency. The use of a two-level NoC is shown to provide an average of 65% sensor data latency improvement versus a flat sensor data NoC structure for a 256-core system.

Categories and Subject Descriptors

C.4 [Performance of Systems] Reliability, availability, and serviceability

General Terms

Performance, Design, Reliability.

Keywords

Many-core, on-chip monitoring, distributed sensor processing.

1. INTRODUCTION

In the next few years many-core processors containing up to 1000 processor cores will become a reality [1]. Due to performance, power and reliability concerns, these massively parallel computing substrates will be required to evaluate an increasing amount of run-time information pertaining to error, thermal, process variation, wear-out, and supply voltage integrity issues, among others. The emergence of three-dimensional (3D) die stacking will further amplify the need for sensor information and corresponding remediation actions [2]. Currently, real-time system responses for multi-cores, including dynamic voltage and

frequency scaling (DVFS), error recovery, and thermal remediation are performed locally and are often isolated within individual cores. As system-on-chips (SoC) scale, both local and global techniques are needed to collect, collate, and use the information obtained from on-chip sensors [3]. These actions require multiple processors for deterministic and low latency sensor data processing.

Our approach to managing sensor data for many-cores involves providing architectural support for distributed sensor data collection and processing and system remediation on a chip-wide basis. In many-core systems, system temperature, voltage droop, processor activity, etc. need to be closely monitored and run-time remediation, such as DVFS, is invoked when necessary [4]. Recent advances in the use of sensor data include the use of processor performance signatures and performance counters to predict voltage droop emergencies and prevent thermal emergencies [3][5]. These advances motivate infrastructure for run-time management based on sensor processing components that share and distribute run-time signature, voltage, thermal, and error information. Unlike previous on-chip monitoring infrastructures [6][7], our architecture includes multiple processing components which are dedicated to sensor data analysis. A hierarchical network-on-chip (NoC), which interconnects the sensor data processors (SDPs), allows for both efficient sensor data collection and inter-SDP communication for shared sensor data. The approach is verified for many-core systems for up to 1024 cores in the core layer. A customized interconnect simulator is used to evaluate the communication infrastructure for sensor data. Additionally, the Graphite many-core simulator [1] is used to evaluate the many-core architecture for a collection of accepted benchmarks. A system-level experiment which examines the global distribution of voltage data and thermal information is performed to show the benefit of using our hierarchical infrastructure.

Experimental results show that our hierarchical sensor data communication infrastructure achieves up to an 80% latency reduction compared to a one-layer infrastructure. The system level benefit of our approach is shown using dynamic frequency scaling (DFS) for thermal management and voltage droop compensation. The results show an average many-core performance improvement of 6% using our hierarchical infrastructure, although higher rates of system temperature and supply voltage change will lead to higher benefits.

The remainder of this paper is organized as follows. Section 2 presents a brief background on many-core systems, 3D stacks and on-chip sensor data systems. Section 3 introduces our many-core sensor data collection and processing infrastructure. Section 4 discusses our experimental approach and experimental results are presented in Section 5. Section 6 concludes the paper.

This work was funded by the Semiconductor Research Corporation under Task 2083.001

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

GLSVLSI'12, May 3-4, 2012, Salt Lake City, UT, USA.

Copyright 2012 ACM 978-1-4503-0012-4/12/05...\$10.00.

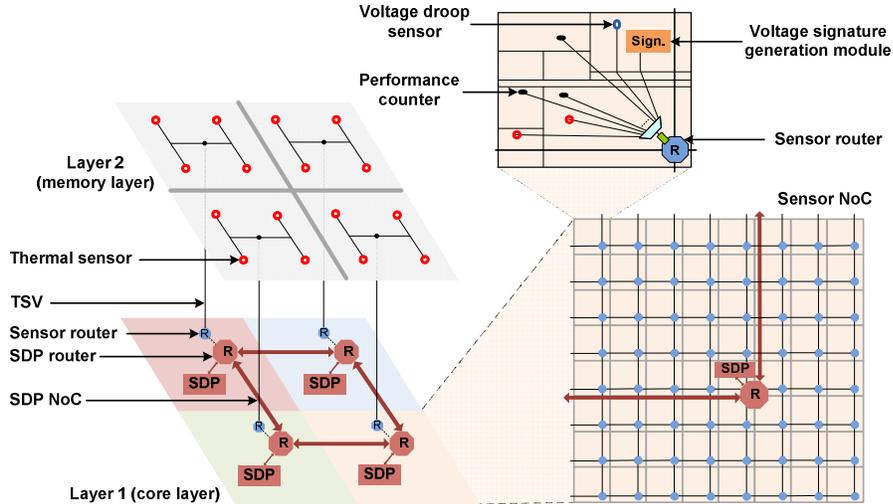


Figure 1. Hierarchical sensor data processing infrastructure for a 256-core system. Two distinct NoCs are shown. The sensor NoC (on the right) connects sensors to SDPs. The SDP NoC (on the left) connects SDPs. Sensors in the memory layer transfer data via TSVs.

2. Background

On-chip sensors are widely used in processors to closely monitor system temperature, performance, supply power fluctuation, and other environmental conditions. For example, the IBM Power7 includes 5 thermal sensors and 31 activity sensors per core [6]. Information from sensors, which is used to perform remediation techniques such as DVFS, voltage droop compensation and error rollback, presents a significant communication and processing workload. In many-cores, the impact of these communication and processing workloads is exacerbated by the highly-distributed nature of hundreds of cores. The dramatic expansion of sensor data necessitates a global and distributed view of sensor data processing.

Numerous remediation approaches based on on-chip sensors have been introduced. DVFS is widely used for processor thermal management in which the system frequency and/or voltage is reduced when a higher-than-threshold temperature is detected [3]. Supply voltage droops pose a threat to multi-core system reliability, thus DFS or voltage boosting needs to be enabled when a significant voltage droop is detected [8]. A recently-developed signature-based voltage droop compensation method detects signatures (a sequence of processor execution events) that are related to significant voltage droops and enables early prediction of incoming droops based on these signatures [8]. System reliability information measured as architectural vulnerability factors (AVF) are monitored at run-time and used to enable redundancy protection (e.g., dual modular redundancy) against soft errors when necessary [9]. Moreover, combinations of on-chip sensor information for multi-core remediation have also been explored [3][9].

Several approaches for on-chip sensor data collection and processing have been introduced. The IBM EnergyScale adaptive energy management approach [6] implements on-chip thermal and critical path sensors and performance counters for each core in an eight-core system. A microcontroller is used for sensor data processing. Intel AMT technology [10] uses a separate communication channel for remote discovery, healing and protection. In Wang, et al. [11], on-chip sensor data is transmitted using the existing NoC for regular inter-processor traffic. However, none of these techniques is suitable for many-core

systems with a large number of distributed on-chip sensors. A previous NoC-based infrastructure [7] for monitoring addressed some of these issues. This system, which is targeted at multi-cores, includes a low-dimensional NoC and up to two microcontrollers for centralized sensor data collection and processing. This earlier interconnect is organized as a flat two-dimensional mesh. No data exchange between the microcontrollers is supported in this infrastructure and the lack of a hierarchical interconnect significantly inhibits the scalability of the interconnect. Although this interconnect is sufficient for multi-cores, the greater throughput demands presented by many-cores motivates a new hierarchical interconnect approach.

The use of 3D stacking technology leads to additional challenges for sensor data collection [2]. Inter-layer communication is facilitated by the use of through silicon vias (TSVs). The total number of TSVs is limited. Most 3D implementations focus on layering memory on top of a processor core layer [2][4]. Although it is expected that multiple stacked core layers will be implemented in the future, this work mainly considers two layer stacks including a memory and a core layer.

3. Distributed Sensor Data Collection and Processing

An overview of the hierarchical infrastructure for distributed sensor data collection and processing is shown in Figure 1 for a two-layer 3D stack many-core implementation. This dedicated interconnect and sensor data processor infrastructure, which only handles sensor data, contains two levels of NoC routers and a series of SDPs. The NoC infrastructure, SDPs, and most of the sensors are implemented in the core layer while thermal sensor data in the memory layer are accessed through TSVs. SDPs can be implemented from available regular cores.

3.1 Hierarchical Sensor Data Interconnect Infrastructure

The sensor data interconnect infrastructure consists of two levels of NoC-style routers. On-chip sensors in each core are connected to a minimalistic packet router through a multiplexer, as shown in the top, right in Figure 1. These routers, called sensor routers, are connected together in a mesh, as shown in the bottom, right in Figure 1. Sensor routers (one per core) send the collected sensor

data to an SDP through the sensor NoC. Data from thermal sensors in the memory layer are also collected by the SDP through the sensor NoC in a slightly different fashion. Adjacent thermal sensors (4 in the example shown in Figure 1, not necessarily the number used in real systems) in the memory layer are connected to a multiplexer which sends its output to a serializer. The thermal data are received in the core layer, de-serialized, sent to a sensor router, and subsequently forwarded to the SDP. This approach only uses one TSV for each vertical connection between the serializer in the memory layer and the de-serializer in the core layer. Ten cycles [12] are required to transmit the 8-bit thermal data from the memory layer to the sensor NoC.

The sensor router implemented in this infrastructure has a small data width (sensor data packet width) and a shallow input buffer (e.g., 24-bit width and 4 flits, much smaller than the 256-bit width and 8-16 flits used in standard NoCs). The sensor router supports data packets with two priority levels using two virtual channels. Packets in the priority channel have higher routing priority than those in the regular channel. Emergency sensor data packets (such as an alert for a significant voltage droop) are transmitted in the priority virtual channel to avoid congestion, thus it has lower latency. The widely-used XY routing algorithm is implemented in the sensor router [13]. Each packet generated by the sensor router includes a time stamp which indicates its generation time. An SDP manages sensor NoC-transferred data from a relatively small number of cores (64 in the example shown at the bottom, right of Figure 1) and is physically placed in the center of these cores to reduce sensor data transmission latency.

As processor counts scale to many-cores, there is a need for the SDPs to quickly share data, as will be shown in the next subsection. This need motivates a second interconnect layer between SDPs to reduce the number of hops needed for transmitting packets between SDPs, as shown in the bottom, left in Figure 1. The communication among SDPs is facilitated by the SDP NoC using very low overhead SDP routers interconnected in a mesh, effectively forming a lightweight higher-level network. The SDP router in our infrastructure is interfaced to both the sensor NoC and the SDP NoC. Sensor data packets from sensor routers are processed by the SDP and, when appropriate (determined by the SDP), sent to adjacent SDPs. Additional SDP router details are explained in Section 3.3.

3.2 Packet Transmission in the SDP NoC

In many cases, sensor data does not need to be shared across multiple cores and can be used locally. For example, for thermal management [4], per-core architectural adaptation is employed to reduce individual core temperatures based on thermal sensor information collected in each core.

However, some recent many-core remediation approaches, such as DVFS in response to hotspot detection and voltage boost in response to voltage droop, require the sharing of sensor information on a global scale, which includes aggregations over regions of different scales and broadcasts. For example, global scale hotspot remediation requires the transfer of thermal information and performance counts to a centralized location [3] and voltage droop recovery requires the broadcast of voltage sensor data. Most NoC-based multi-core and many-core systems do not have broadcast support in the NoC, since the NoC is mainly used for accessing shared memory.

In our system for the hotspot case, thermal sensor data packets are sent from all SDPs to a central SDP (aggregation at the chip scale

[3]) which determines frequency change decisions. For the voltage droop case, voltage sensor data packets are broadcast among all SDPs. Voltage droop problems affect every core in a many-core system since they are on the same power grid [8]. Thus, a dangerous voltage droop detected in one core should be known by all other cores so that remediation, such as voltage boost or frequency reduction, can be enabled globally [5].

The SDP NoC facilitates both of these traffic patterns. The XY routing algorithm is implemented in the SDP router. Hotspot traffic is supported by this routing algorithm. The SDP router supports packet broadcast using an accepted approach [14]. A packet is first sent vertically to all nodes (along the Y axis) across the mesh. Then, all the nodes that currently have the packet send it out horizontally (to the left and right along the X axis). In a mesh network of size $n \times n$, the position of a router is represented as (x, y) in which $0 \leq x, y < n$ and x, y are both integers. There are two scenarios in which new packets need to be generated. At the source SDP router, up to four new packets are generated and sent to router $(x, 0)$, $(x, n-1)$, $(0, y)$ and $(n-1, y)$. At routers on the same row as the source router, new broadcast packets are generated and sent to router $(0, y)$ and $(n-1, y)$. These new packets follow the XY routing algorithm to their destinations.

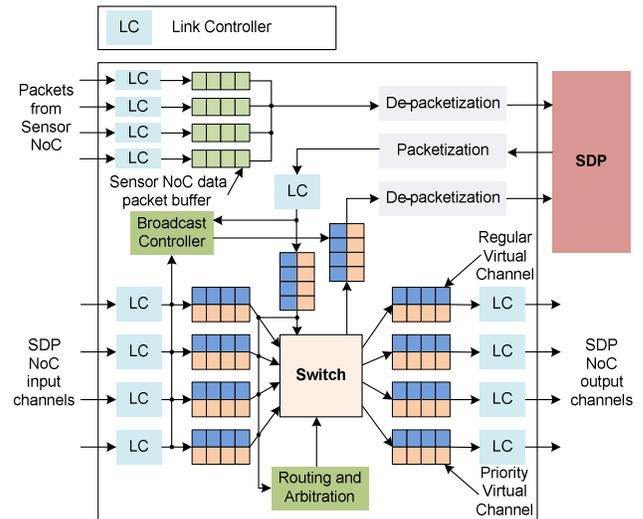


Figure 2. SDP router structure. The structure is simplified from more standard NoCs. Each packet has minimal bit width (24 bit maximum) and storage buffers are shallow (6 flits)

3.3 SDP Router Design

The structure of the SDP router in our infrastructure is shown in Figure 2. The link controller in this figure is responsible for controlling when packets can be sent to/from the buffer based on the usage of the buffer. The SDP router is interfaced to both the sensor NoC and the SDP NoC using shallow buffers (about 6 flits). An SDP write path to the sensor NoC is unneeded so only an input path is provided. Sensor data is extracted from these packets in the de-packetization module and sent to the SDP. Outdated regular sensor data packets, as calculated from the time stamp in the packet, can be discarded (e.g., thermal sensor data packet from the previous sampling period). A crossbar is used to implement the switch.

Similar to the sensor router structure, two virtual channels for regular (e.g., thermal sensor data) and priority packets (e.g. high voltage droop alerts) are implemented in each input and output

buffer. The XY routing algorithm is used in the routing and arbitration module. This structure is enhanced with a broadcast controller for performing broadcasts. When a broadcast packet is received at any router buffer, it is sent to the output buffer for the SDP. At the same time, the broadcast control module decides whether new packets need to be generated, based on the method explained in Section 3.2. These new packets have the same sensor data as the original packet but with different destinations.

4. Experimental Approach

A series of simulation and synthesis evaluations were performed to show the benefit of using our infrastructure for 256, 512 and 1024 many-core systems with one memory layer and one core layer in a two layer stack. Two specific sensor data interconnect approaches are considered, the hierarchical approach shown in Figure 1 and a flat sensor data interconnect that consists only of sensor routers (similar to the infrastructure shown in Figure 1 but *without* the SDP NoC). A packet transmitted between two neighboring SDPs in the flat sensor NoC infrastructure needs to go through several sensor routers. The Popnet simulator [15] is heavily modified to model both the new hierarchical infrastructure and the flat sensor NoC infrastructure for many-core systems.

To estimate the overhead of our infrastructure, synthesizable hardware models of the sensor NoC and SDP routers were developed. The hardware models were synthesized by Synopsys Design Compiler using a 45nm standard cell library [16]. The system-level effect of a many-core system with a core layer and a separate DRAM (memory) layer was modeled using the Graphite many-core simulator [1] with a previously-determined memory access latency number for a 3D stack [2]. The performance calculation module in Graphite has been modified to accommodate run-time frequency changes and to report the overall performance of the system with dynamic frequency scaling. The system frequency is set to 1 GHz. The temperature effect of stacking a DRAM layer on top of the core layer is estimated with a highly-accurate many-core temperature estimation method based on the power consumption of all cores. It is assumed that the heat sink is below the core layer. Thus, the temperature in the core layer is proportional to the power consumption in both the core layer and the DRAM layer [2].

The 128-core architecture used in the system level experiment is scaled up from an 8-core UltraSPARC T1 architecture consuming 115mm² using 90nm technology [4]. The total area of the 128 core system is estimated to be 460mm² using 45nm technology. The power modeling method used in [4] is adopted in our experimentation and scaled to 45nm technology. A maximum value of 5.24 W/core was determined for a 45nm 128-core system. The DRAM layer has the same area as the core layer and hosts 2 GB memory [2]. The DRAM power consumption is set to 1 W/GB [17].

5. Experimental Results

A series of simulations was first performed to find the optimal number of cores per SDP and to show the benefit of using our hierarchical infrastructure versus a flat sensor NoC infrastructure for sensor data distribution. A system-level experiment was then performed for the many-core system using our infrastructure for thermal and voltage droop sensor data transmission.

5.1 On-chip Sensor Setup

In a series of simulations, sensor data from thermal sensors, processor performance counters, and voltage droop sensors were

considered based on previously-reported instantiation and sampling rates in multi-core systems. Eight thermal sensors [7], 18 performance counters [3], a voltage droop signature capturing structure and a voltage droop sensor [8] are used in each core. The total number of thermal sensors in the DRAM layer is 128 (one thermal sensor per 128Mb DRAM [18]). Hardware synthesis indicates that our infrastructure can run at 1 GHz (both the sensor NoC and the SDP NoC).

Thermal data, performance counter data and voltage droop signature data are transmitted in the regular sensor NoC channel. The thermal sensor data injection rate per sensor is based on the maximum temperature rise rate of 10°C/ms [19] and the thermal sensor resolution of 0.1°C [5], which leads to a sample period of 10,000 cycles (1/10,000 cycle injection rate). The performance counter injection rate is 1/3,000,000 cycles [3]. A voltage droop signature injection rate of 1/4,000 cycles is used [8]. Thus, the sensor router regular channel data injection rate is 1/947 cycle based on 8 thermal sensors, 18 performance counters and 1 signature capturing structure. To reduce the total number of TSVs used for transmitting sensor data from the DRAM layer, every 8 thermal sensors in the DRAM are connected to a sensor router. Thus, there are 16 sensor routers that also receive thermal information from sensors in the DRAM layer. These sensor routers are evenly distributed in the core layer. The sensor packet injection rate at these routers is 1/539 cycle based on 16 thermal sensors, 18 performance counters and 1 signature capturing structure. The voltage droop sensor sends out an alert when a dangerous voltage droop happens and this packet is transmitted in the priority channel. A very aggressive injection rate of 1/108 is used for the voltage droop sensor [8].

The SDP NoC traffic includes performance counter and voltage droop broadcast traffic. The performance counter data in the n cores managed by a SDP are sent to the central SDP. Every signature used in the core layer is broadcast to all SDPs. Voltage droop alerts are also broadcast in the SDP NoC using the priority channel. The sensor NoC and SDP NoC data widths are set to 24 bits [7]. The regular packet size for thermal sensor and performance counter values is 1 flit [9] while the signature information requires 3 flits [8]. The buffer sizes in the sensor NoC and the SDP NoC are set to 6 flits.

5.2 Core to SDP Ratio Experiment

In the first experiment, we simulated the infrastructure introduced in this paper with varying numbers of cores per SDP. The total number of SDPs decreases as the number of cores per SDP increases. The latency and hardware cost (including wire area) results using varying cores-per-SDP ratios for 256, 512 and 1024 core systems are shown in Table 1. The packet latency numbers shown in this table are for packets transmitted using both the sensor NoC and the SDP NoC. The TSV latency has been included in this experiment.

As shown in Table 1, the average sensor NoC latency (sensor-to-SDP latency) increases as the cores-per-SDP ratio increases since there are more sensors connected to the each SDP through the sensor NoC. The SDP NoC latency (SDP-to-SDP latency) decreases as the ratio increases since the size of the SDP NoC is reduced. The minimum overall latency is located in the middle of the extremes, although the latency differences are relatively small. The SDP NoC hardware cost decreases as the cores-per-SDP ratio increases (fewer SDP routers and shorter interconnect).

Table 1. Average latency (in cycles) and hardware costs associated with varying core count per SDP

Core num.	SDP num.	Core/SDP ratio	Sensor NoC lat.	SDP NoC lat.	Total lat.	SDP+sensor NoC to sensor NoC-only HW increase (%)
256	32	8	7.09	18.83	25.92	24.98
	16	16	9.35	13.62	22.97	14.20
	8	32	12.25	11.01	23.26	7.94
	4	64	17.31	7.72	25.03	4.19
	2	128	24.14	5.25	29.39	1.72
512	64	8	7.09	24.39	31.48	25.93
	32	16	9.35	19.34	28.69	15.09
	16	32	12.25	13.37	25.62	8.78
	8	64	17.31	11.42	28.73	4.99
	4	128	24.14	7.88	32.02	2.65
1024	128	8	7.09	39.40	46.49	26.22
	64	16	9.35	26.01	35.26	15.46
	32	32	12.25	20.87	33.12	9.20
	16	64	17.31	15.48	32.79	5.46
	8	128	24.14	11.84	35.98	3.15

A cores-to-SDP ratio of 64 is used in the following experiments since it provides low latency and the highest capacity for sensor packets while requiring moderate hardware cost (less than 6% increase versus sensor NoC-only in the 1024 core system). The hardware cost of both NoCs together is less than 1.5% of the overall many-core hardware area for all the cases, since the data width and buffer size for both NoCs are small and a simplified router structure (versus standard NoC routers) is used, as explained in Section 3.1 and 3.3.

Table 2. Average latency (in cycles) comparison

Core and SDP num.	Latency type	Flat sensor NoC	Our method	Latency reduction w.r.t. flat sensor NoC (%)
256 core (4 SDP)	Inter-SDP	45.43	7.72	83.01
	Total	62.82	25.03	60.16
576 core (8 SDP)	Inter-SDP	67.57	11.42	83.10
	Total	84.96	28.73	66.18
1024 core (16 SDP)	Inter-SDP	90.36	15.48	82.87
	Total	107.75	32.79	69.57

5.3 Comparison against the Flat Sensor NoC Infrastructure

In this experiment, the hierarchical approach described in this paper is compared against a flat one-layer (sensor NoC-only) sensor data interconnect. As shown in Table 2, our hierarchical infrastructure achieves 60%, 66% and 70% total latency (SDP NoC latency + sensor NoC latency) reduction versus the flat sensor NoC infrastructure for 256, 512, and 1024 cores, respectively. The sensor NoC latency in both infrastructures barely changes as the total core number increases since each SDP manages the same number of sensors as the total core count increases.

The inter-SDP sensor packet transmission latency makes the difference. The SDP NoC latency in our infrastructure is over 80% lower versus the inter-SDP latency in the flat sensor NoC infrastructure. As the total core number increases from 256 to 1024, the difference becomes larger. As explained in Section 3, SDPs are directly connected in our hierarchical infrastructure while inter-SDP packets in the flat sensor NoC infrastructure needs to go through numerous sensor routers (at least 8 sensor routers for neighboring SDPs in this experiment).

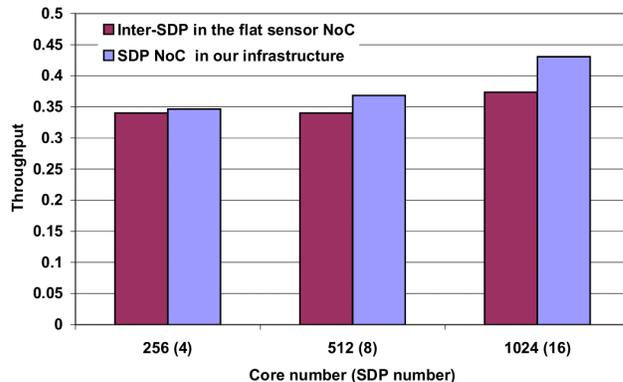


Figure 3. Throughput (packet/SDP router/cycle) comparison

The throughput of the SDP NoC in 256, 512 and 1024 core systems is shown in Figure 3. For comparison, the throughput of the inter-SDP traffic in the flat sensor NoC infrastructure is also shown. The sensor NoC and the vertical communication structure are not included in this simulation since they yield the same throughput in both infrastructures.

The throughput of the SDP NoC is defined in Equation (1) [20]. P_{total} is the total number of packets received during the simulation, N is the number of routers in the SDP NoC and C is the average number of cycles to route all the packets, which is from the simulation.

$$Throughput = \frac{P_{total}}{N \times C} \quad (1)$$

Figure 3 shows that the SDP NoC has higher throughput for all simulated systems. The difference increases as the number of cores increases since the broadcast requirements of the sensor data negatively impacts the sensor NoC-only case.

5.4 Using DFS for Thermal Management and Voltage Droop Compensation

In a system level experiment, a 2-layer 128-core system with integrated voltage droop sensors and thermal sensors is simulated. A voltage droop alert sent by the voltage droop sensor is broadcast to all SDPs. The frequency of all cores is reduced by half during a voltage droop [5]. Thermal sensor data is processed in the local SDP only. The system frequency is reduced by half when the temperature is over 85°C [4].

The sensor setup includes 1 voltage droop sensor per core [5] and 8 thermal sensors per core. The DRAM layer with 2 GB capacity has 128 thermal sensors, as explained in Section 5.1. No voltage droop sensors are used in the memory layer. The injection rate of the voltage droop sensor and thermal sensor is 1/108 and 1/10,000 cycle respectively, as explained in Section 5.1. In this experiment, each SDP manages sensors in 32 cores, thus 4 SDPs are used in this 128-core system. The hardware cost of adding the SDP NoC is less than 7% of the overall monitoring system hardware cost (sensor NoC, SDP, and SDP NoC). The overall hardware cost of our infrastructure is only 0.94% of the total chip (core layer).

Three cases are considered in this experiment.

- 1) Case 1: DFS for thermal management only. Without on-chip voltage droop sensors, the system voltage is conservatively set to 1.1V [7].
- 2) Case 2: DFS for thermal management and voltage droop compensation using the flat sensor NoC.

3) Case 3: DFS for thermal management and voltage droop compensation using the hierarchical infrastructure introduced in this paper.

The latency of sensor data transmission is simulated using the modified Popnet simulator, as described in Section 4. This simulation shows that the latencies for transmitting voltage droop sensor information using the flat sensor NoC infrastructure and our infrastructure are 62 cycles and 21 cycles, respectively. Using the system voltage calculation method described in [7] and scaling it to 45nm technology, the system voltage is set to 1.02V and 1V for case 2 and 3 respectively. Given the same processor activity, the maximum temperature difference from these two voltages is close to 6°C using the temperature estimation method described in Section 4. Case 1 is used as a baseline case for comparison.

Table 3. Results of the system level experiment using DFS for thermal management and voltage droop compensation

Benchmark	Perf. (billion cycles)			Benefit (%)	
	Case 1	Case 2	Case 3	Case 2	Case 3
LU (non-contig)	89.33	81.41	79.46	8.87	11.05
Ocean (non-contig)	3.49	3.26	3.25	6.56	6.74
LU(contig)	24.23	23.70	22.21	2.20	8.32
Ocean(contig)	2.76	2.54	2.51	7.79	9.09
Radix	9.78	9.29	9.08	5.03	7.18
FFT	115.14	115.06	114.73	0.07	0.36
Cholesky	189.66	185.05	182.28	2.43	3.89
Radiosity	121.42	114.71	111.28	5.53	8.35

The results are shown in Table 3. The performance is represented as the total cycles for the 128 core system (sum of execution time in each core) to finish the benchmark. The execution time difference shown in Table 3 for case 2 and 3 are with respect to the execution time in case 1. The system using our infrastructure (case 3) achieves an average 6.8% performance benefit compared to the system without on-chip voltage droop sensors (case 1). The performance benefit is up to 6% higher compared to a system with the flat sensor NoC infrastructure (case 2). Case 3 uses the lowest supply voltage, which leads to the lower system temperature and less DFS enable time using our infrastructure. The benefit is small with the FFT benchmark since the system temperature is almost always below the threshold.

6. Conclusion

A dedicated infrastructure for distributed sensor data collection and processing for many-core systems is introduced. This infrastructure features a hierarchical NoC that supports two types of sensor data traffic. Our infrastructure achieves more than 50% latency reduction versus a flat NoC infrastructure in many-core systems. The system level performance benefit of using our infrastructure is up to 6% versus a nominal system.

7. References

- [1] J. Miller, et al., "Graphite: A distributed parallel simulator for multicores," in *Proc. Int'l Symp. on High Performance Computer Architecture*, pp. 1-12, Jan. 2010.
- [2] G. Loh, "3D-stacked memory architectures for multi-core processors," in *Proc. Int'l Symp. on Computer Architecture*, pp. 453-464, Jun. 2008.
- [3] R. Jayaseelan, et. al., "A hybrid local-global approach for multi-core thermal management," in *Proc. Int'l Conf. on Computer-Aided Design*, pp. 314-320, Nov. 2009.
- [4] A. Coskun, et al., "Dynamic thermal management in 3D multicore architectures," in *Proc. Design, Automation & Test in Europe Conf.*, pp. 1410-1415, Apr. 2009.
- [5] J. Zhao, et al., "Thermal-aware voltage droop compensation for multi-core architectures," in *Proc. Great Lakes Symp. on VLSI*, pp. 335-340, May 2010.
- [6] M. Floyd, et al., "Adaptive Energy Management Features of the POWER7 Processor," [online]. Available: http://www.research.ibm.com/people/l/lefurgy/Publications/hotchips22_power7.pdf.
- [7] S. Madduri, et al., "A monitor interconnect and support subsystem for multicore processors," in *Proc. IEEE Design Automation & Test in Europe Conf.*, pp. 761-766, Apr. 2009.
- [8] V. Reddi, et. al. "Voltage emergency prediction: A signature-based approach to reducing voltage emergencies," in *Proc. Int'l Symp. on High-Performance Computer Architecture*, pp. 18-27, Feb. 2009.
- [9] R. Vadlamani, et al., "Multicore soft error rate stabilization using adaptive dual modular redundancy," in *Proc. Design Automation and Test in Europe Conf.*, pp. 27-32, Mar. 2010.
- [10] "Intel Active Management Technology," [online]. Available: <http://www.intel.com/technology/platform-technology/intel-amt/>
- [11] Y. Wang, et al., "Performance evaluation of on-chip sensor network (SENoC) in MPSoC," in *Proc. Int'l Conf. on Green Circuits and Systems*, pp. 323-327, Jun. 2010.
- [12] S. Pasricha, "Exploring serial vertical interconnects for 3D ICs," in *Proc. Design Automation Conf.*, pp. 581-586, Jul. 2009.
- [13] M. Yang, et. al., "Incremental design of scalable interconnection networks using basic building blocks," in *Proc. IEEE Symp. on Parallel and Distributed Processing*, pp. 252-259, 25-28, Oct. 1995.
- [14] E. Modiano, et. al., "Efficient algorithms for performing packet broadcasts in a mesh network," in *IEEE Trans. on Networking*, vol. 4, no. 4, pp. 639-648, Aug. 1996.
- [15] L. Shang, et. al., "Dynamic voltage scaling with links for power optimization of interconnection networks," in *Proc. Int'l Symp. on High-Performance Computer Architecture*, pp. 91-102, Feb. 2003.
- [16] The Nangate 45nm Open Cell Library [online]. Available: <http://www.nangate.com>
- [17] Samsung Green DDR3, [online]. Available: <http://www.samsung.com/global/business/semiconductor/Greenmemory>
- [18] C. Kim, et al., "CMOS temperature sensor with ring oscillator for mobile DRAM self-refresh control," in *Proc. Int'l Symp. on Circuits and Systems*, pp. 3094-3097, May 2008.
- [19] G. Zhao, et al., "Processor frequency assignment in 3D MPSoCs under thermal constraints by polynomial programming," in *Proc. Asia Pacific Conf. on Circuits and Systems*, pp. 1668-1671, Nov. 2008.
- [20] A. Weldezion, et al., "Scalability of network-on-chip communication architecture for 3-D meshes," in *Proc. Int'l Symp. on Networks-on-Chip*, pp. 114-123, May 2009.