

3-2012

# Location of Processor Allocator and Job Scheduler and its Impact on CMP Performance

Dawid Zydek

Idaho State University, zydedawi@isu.edu

Grzegorz Chmaj

University of Nevada, Las Vegas, chmajg@unlv.nevada.edu

Alaa Shawky

Idaho State University, mohaalaa@isu.edu

Henry Selvaraj

University of Nevada, Las Vegas, henry.selvaraj@unlv.edu

Follow this and additional works at: [https://digitalscholarship.unlv.edu/ece\\_fac\\_articles](https://digitalscholarship.unlv.edu/ece_fac_articles)



Part of the [Computer and Systems Architecture Commons](#), and the [Electrical and Computer Engineering Commons](#)

## Repository Citation

Zydek, D., Chmaj, G., Shawky, A., Selvaraj, H. (2012). Location of Processor Allocator and Job Scheduler and its Impact on CMP Performance. *International Journal of Electronics and Telecommunications*, 58(1), 9-14.  
[https://digitalscholarship.unlv.edu/ece\\_fac\\_articles/843](https://digitalscholarship.unlv.edu/ece_fac_articles/843)

This Article is brought to you for free and open access by the Electrical & Computer Engineering at Digital Scholarship@UNLV. It has been accepted for inclusion in Electrical and Computer Engineering Faculty Publications by an authorized administrator of Digital Scholarship@UNLV. For more information, please contact [digitalscholarship@unlv.edu](mailto:digitalscholarship@unlv.edu).

# Location of Processor Allocator and Job Scheduler and Its Impact on CMP Performance

Dawid Zydek, Grzegorz Chmaj, Alaa Shawky, and Henry Selvaraj

**Abstract**—High Performance Computing (HPC) architectures are being developed continually with an aim of achieving exascale capability by 2020. Processors that are being developed and used as nodes in HPC systems are Chip Multiprocessors (CMPs) with a number of cores. In this paper, we continue our effort towards a better processor allocation process. The Processor Allocator (PA) and Job Scheduler (JS) proposed and implemented in our previous works are explored in the context of its best location on the chip. We propose a system, where all locations on a chip can be analyzed, considering energy used by Network-on-Chip (NoC), PA and JS, and processing elements. We present energy models for the researched CMP components, mathematical model of the system, and experimentation system. Based on experimental results, proper placement of PA and JS on a chip can provide up to 45% NoC energy savings.

**Keywords**—CMP, PA, JS, energy, assignment.

## I. INTRODUCTION

CHIP MULTIPROCESSOR (CMP) is a single computing unit with two or more Processing Elements (PEs) called cores. The cores are integrated on a single die. It delivers better latency and bandwidth performance, but such aspects as energy and area become crucial. Since their first appearance in 2005, CMPs have evolved from 2-core architectures to 32-core processors that are available in the market today [1]. Moreover, current technology allows designing CMPs with many more cores, e.g. Intel Teraflop (80 cores) or TILE-Gx (100 cores).

Multicore CMPs are characterized by tiled architecture, where area of a chip is divided into tiles (Fig. 1). Besides PE and cache memory, each tile contains networking interface and router (R) that ensures communication among PEs. Routers are connected by physical channels implemented across a chip that forms Network-on-Chip (NoC) [2].

NoCs and PEs are subjects of intense research [2]–[4]. Among several NoC architectures, NoCs with low-dimensional topologies represent higher throughput and lower latency in comparison to high-dimensional networks. It favors topologies like 2D-Mesh and 2D-Torus. Both topologies match very closely with the physical layout of the die and contain many redundant paths, that makes them very attractive for current and future CMPs [5], [6]. In this paper, these two NoC topologies are considered.

Design of the NoC and the architecture of PEs have significant impact on CMP performance. However, even with the

Dawid Zydek and Alaa Shawky are with the Department of Electrical Engineering, Idaho State University, USA (e-mails: zydedawi@isu.edu; mohaalaa@isu.edu).

Grzegorz Chmaj and Henry Selvaraj are with the Department of Electrical and Computer Engineering, University of Nevada, Las Vegas, USA (e-mails: chmajg@unlv.nevada.edu; selvaraj@unlv.nevada.edu).

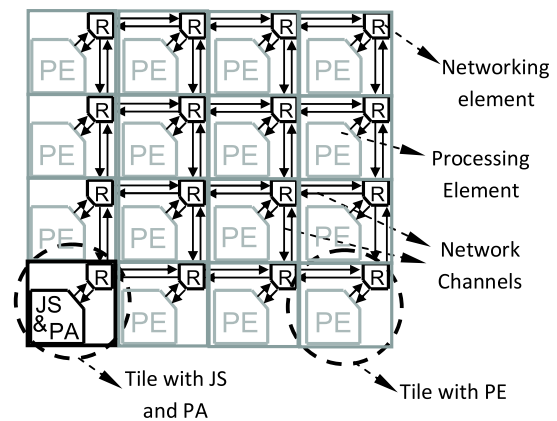


Fig. 1. Tiled CMP (4 × 4 2D-mesh) with integrated a job scheduler and processor allocator.

best organized CMP, poor utilization of many available cores may lead to efficiency degradation. Thus, effective use of cores available on a CMP is also an important factor. On-chip PEs can be used to run a single-task job or they can serve to execute multi-task job where all tasks may be done in parallel using many PEs (parallel processing). The efficient use of the PEs in a CMP is supervised by the processor management system, which contains two components: Processor Allocator (PA) and Job Scheduler (JS). The JS is responsible for job scheduling the selection of the job to be executed next. In this paper as job scheduling policy we consider the FCFS (First Come First Served) fashion. The PA is in charge of processor allocation selecting a set of PEs required for a given job. The processor allocation has to be fast enough to meet high performance offered by CMPs. This led to the idea of hardware implementation of PA and JS, and integrating them together with PEs on one die (Fig. 1) [7].

Internal hardware design of PAs may vary based on processor allocation technique and algorithm used [7], [8]. There are two major categories of processor allocation: contiguous and non-contiguous. In non-contiguous approach, tasks of job can be executed on multiple disjoint smaller subgrids. The PEs handling the tasks do not have to be physically adjacent. In contiguous allocation strategy, the PEs allocated to job are physically adjacent and have the same topology like NoC. In this paper we use contiguous processor allocation strategy since it is more effective for CMPs [7], [9].

A lot of research has been done to increase the efficiency of NoC-based CMPs with processor allocator system. NoCs have been studied in [2], [3], [10]–[12]. Processor allocation algorithms are discussed in [7], [8], [13], [14]. A hardware

implementation of PA and JS is described in [7]. For the proposed CMP with embedded PA and JS, an energy model is described in [15] and the performance is evaluated in [9]. In this paper, we present the problem of PA and JS location on the chip and its impact on the NoC and performance of the allocation process. We consider many locations across the chip and for each location the energy and traffic balance results are examined and compared. As a simulation environment, we use the system described in [9]. Some example implementations of this kind of systems are shown in [16]–[18]. Simulated configurations are presented and described, together with their results.

The rest of the paper is organized as follows: Section II describes the presented problem. Experimentation system and examples of experiments are shown and discussed in Section III, while closing remarks are in Section IV.

## II. PROBLEM DESCRIPTION

In CMPs, PEs and on-chip network are significantly closer than in off-chip multiprocessor system. It ensures better latency and bandwidth performance, but such properties like power, area and cost restrictions deserve closer attention. All components of the chip have impact on its properties, thus PEs, PA with JS, and NoC need to be carefully designed and implemented. A physical layout of the considered CMP is presented in Fig. 1. A detailed description can be found in [7]. The chip area is divided into tiles, that ensures scalability and effective use of resources available on the chip. Each tile contains networking elements (router, networking interface, network channels) and PEs (processor, cache memory, etc.). Communication among tiles is executed by sending messages over the NoC using routers. We consider a homogenous architecture, where hardware design and computational power of all PEs in CMP are the same. One of the tiles available on a chip does not contain a PE – it has a hardware version of PA and JS. Both PA and JS support the efficient utilization of PEs in the CMP.

CMPs are designed to process jobs in the most efficient and fast way. In our system, a job may contain one or many tasks that are adjacent to each other. A job has a shape that is a subgrid of the NoC topology, and it is described by the size of the subgrid it requires (Fig. 2). Each PE may process



Fig. 2. A job b that contains 6 tasks and mesh T(3,3).

only one task in the same time so, for jobs containing more than one task, more PEs are needed, e.g. for the job from Fig. 2, six adjacent PEs are needed and their shape must be as illustrated in the figure. Once a job is allocated to PEs, it runs until completion.

### A. Simulation Process

A queue with jobs for processing is randomly generated using discrete uniform distribution. The queue is processed in FCFS fashion by JS. The simulation starts when JS takes the first job from the queue for allocation. The scheduled job is moved to a PA, where it is assigned to available PEs according to allocation algorithm. We use the best allocation techniques: IFF algorithm for 2D-Mesh [7] and BMAT algorithm for 2D-Torus [8]. Once the PA finds available PEs to accommodate the job, the PA sends an allocation message to PEs to reserve them for the job. The jobs are allocated in such a manner that they cannot overlap with each other. If there is no free PEs, the PA waits until another job will release some PEs. After a job is executed, PEs send a release message to the PA, which updates the status of processors. All messages in the system are sent by implemented NoC. The one researched in this paper has the width of NoC channels equal to 32 bits. Thus, one flit has a width of 32 bits and for simplicity we assumed that one packet contains one flit. We assumed as well, that allocation and release messages take one flit, e.g. if a job requires 6 processors, 6 flits have to be sent from a PA to all 4 PEs assigned to the job.

### B. Mathematical Description

#### Indices:

$v, w = 1, 2, \dots, V$	PEs
$M$	PA&JS
$b = 1, 2, \dots, B$	job to process
$s = 1, 2, \dots, S$	sizes of jobs
$t = 1, 2, \dots, T$	time slots

#### Binary variables:

$q_{bs} = 1$	when job $b$ has horizontal size $s$ or less, 0 otherwise (binary)
$r_{bs} = 1$	when job $b$ has vertical size $s$ or less, 0 otherwise (binary)
$y_{bMvt} = 1$	when job $b$ is sent from PA to PE $v$ in time slot $t$ , 0 otherwise (binary)
$x_{bvt} = 1$	when job $b$ is computed at PE $v$ in time slot $t$
$i, j = 1, 2, \dots, N$	indices for position of PE
$g_{vij} = 1$	when PE $v$ resides at position $i, j$ ( $i = \text{horizontal}, j = \text{vertical}$ ) in mesh or torus structure

#### Constants:

$E_{bit}^{v,w}$	energy consumption to send one bit from $v$ to $w$
$W$	word length
$X, Y$	size of mesh/torus (horizontal/vertical)

#### Criterion function:

$$\underset{y}{\text{minimize}} F = 2W \sum_w \sum_b \sum_t y_{bMwt} E_{bit}^{M,w} \sum_s q_{bs} \sum_s r_{bs}$$

**Constraints:**

All jobs have to be computed:

$$\sum_b \sum_v x_{bv} = B \quad (1)$$

Each job is computed once:

$$\sum_v x_{bv} = 1 \quad b = 1, 2, \dots, B \quad (2)$$

PEs do not exchange data packets between each other:

$$\sum_b \sum_v \sum_w \sum_t y_{bvwt} = 0 \quad v \neq w \neq M \quad (3)$$

Job is allocated to PE which is not occupied:

$$\sum_b x_{bvt} = 1 \quad t = 1, 2, \dots, T; v = 1, 2, \dots, V \quad (4)$$

**Mesh specific constraints:**

Job is allocated to adjacent PEs, job must not overlap mesh:

$$\begin{aligned} \sum_v \sum_e \sum_f x_{bvvt} g_{v(i+e)(j+f)} &= \sum_s q_{bs} \sum_s r_{bs} \\ 1 \leq t \leq T, b &= 1, 2, \dots, B \\ 0 \leq e < \sum_s q_{bs}, 0 \leq f < \sum_s r_{bs} \\ 1 \leq i < X - \sum_s q_{bs}, 1 \leq j < Y - \sum_s r_{bs} \end{aligned} \quad (5)$$

**Torus specific constraints:**

Job is allocated to adjacent PEs, job can overlap torus:

$$\begin{aligned} \sum_v \sum_e \sum_f x_{bvvt} g_{v(i+e)\%X(j+f)\%Y} &= \sum_s q_{bs} \sum_s r_{bs} \\ 1 \leq t \leq T, b &= 1, 2, \dots, B \\ 0 \leq e < \sum_s q_{bs}, 0 \leq f < \sum_s r_{bs} \\ 1 \leq i \leq X, 1 \leq j \leq Y \end{aligned} \quad (6)$$

, where % symbol denotes modulo division.

**C. Energy Model**

In the presented problem, we evaluate energy consumed by PA and JS, NoC, and PEs. Since PA and JS are implemented in hardware, we can calculate the exact amount of energy consumption per cycle. The PA and JS were synthesized using Alteras Stratix III family device EP3SL150F780C2 in [7] and energy was estimated in [15]. The energy consumed in a cycle is expressed by formula:

$$E_c^{PA,JS} = P \frac{1}{F_{max}} [\mu J] \quad (7)$$

, where  $P$  is the average power dissipation and  $F_{max}$  is the average maximum frequency of  $f_{max}$  at 0 [°C] and 85 [°C] in [MHz], [7], [19]. We investigate NoC architectures with:

- 1) 2D-Mesh and 2D-Torus topologies,
- 2) Virtual-Channel (VC) and Express Virtual-Channel (EVC) flow controls,
- 3) Dimensional Order Routing with Load Balance extension (DOR-LB) [7].

Each NoC node consists of a EVC router (R in Fig. 1), A packet traversing from a node (tile)  $v$  to neighboring tile  $w$  (it is one NoC channel or 1 hop) needs to be processed by the EVC router where next destination node is selected, and it needs to travers NoC Channel. The average energy consumption in pJ for sending one bit of data from tile  $v$  to tile  $w$  is expressed by:

- For 2D-Mesh:

$$E_{bit}^{v,w} = 0.98(N_{hops}^{VC} + 1) + 0.23N_{hops}^{EVC} + 0.57N_{hops}, \quad (8)$$

- For 2D-Torus:

$$E_{bit}^{v,w} = 0.98(N_{hops}^{VC} + 1) + 0.23N_{hops}^{EVC} + 0.75N_{hops}, \quad (9)$$

, where  $N_{hops}^{VC}$  is the number of regular VCs traversed by a packet between tile  $v$  and  $w$ ,  $N_{hops}^{EVC}$  is the number of EVCs traversed by a packet between tile  $v$  and  $w$ , and  $N_{hops}$  is the number of physical channels (number of EVCs + number of VCs - 1) traversed by a packet between tile  $v$  and  $w$ . The values 0.98 and 0.57 are obtained based on hardware implementation of NoC on an FPGA device [15].

We consider the system built with Intel Core i5-660 processors having a 3.6 GHz clock. These units include two physical cores inside. We treat one i5-660 chip as one PE. According to Intel technical specifications [20] i5-660 has Thermal Design Power (TDP) of 73 [W]. Computing power expressed in GFLOPs equals to 29. We use TDP as the operating power of cluster processors to give a good estimate of energy consumption. We convert the TDP into energy consumed in a cycle  $E_c$  according to formula:

$$E_c^{PE} = TDP \frac{1}{F_{max}} [\mu J] \quad (10)$$

, where  $F_{max}$  is the maximum frequency of Intel Core i5-660 processor in [MHz].

**D. Investigated Characteristic**

The PA and JS can be placed in any node of the NoC, as shown in Fig. 3. In [7], [9], [15], [19], the behavior of

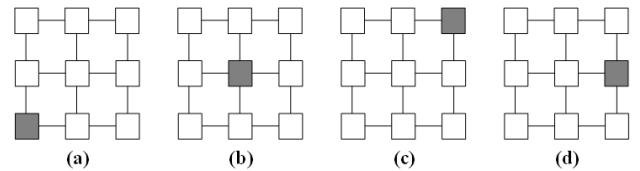


Fig. 3. 2D-Mesh CMP and different location of the PA: (a) (0, 0); (b) (1, 1); (c) (2, 2); (d) (2, 1).

the CMPs was explored with the PA and JS located in the node (0, 0) (Fig. 1 and 3a). In this work, we consider other locations of the PA on the chip and we investigate which location provides the best energy characteristic.

**III. EXPERIMENTATION SYSTEM****A. Structure of the System**

A physical structure of the system is based on the concept presented in Fig. 1. The logical scheme of the conducted experiments is presented in Fig. 4. The problem parameters:

- $P_1$  – horizontal size of the mesh/torus,
- $P_2$  – vertical size of the mesh/torus,
- $P_3$  – topology (2D-Mesh or 2D-Torus),
- $P_4$  – flow control (VC or EVC),

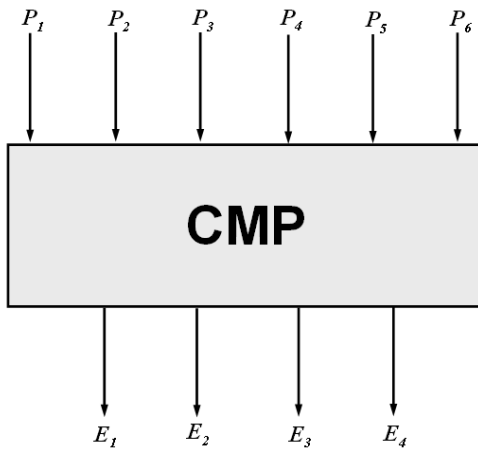


Fig. 4. Block-diagram of the simulated system as input-output system.

- $P_5$  – routing algorithm (DOR, Valiant, DOR-LB, Valiant-LB or Adaptive),
- $P_6$  – locations of the PA (coordinates).

The output parameters are:

- $E_1$  – virtual channel count,
- $E_2$  – express virtual channel count,
- $E_3$  – total virtual channel count,
- $E_4$  – total express virtual channel count.

A detailed description of the output parameters can be found in [9].

### B. Experiments

Using the presented system, we conducted several experiments. We researched CMPs with mesh/torus sizes:  $4 \times 4$ ,  $5 \times 4$ ,  $5 \times 5$ ,  $6 \times 6$ ,  $8 \times 8$ ,  $10 \times 10$ , and  $15 \times 10$ . We have employed the best processor allocation algorithms for CMP, i.e. IFF for 2D-Mesh and BMAT for 2D-Torus. As a routing algorithm, we used DOR-LB algorithm that is most energy efficient and ensures very good load balance. In the experiments, for the same queue with jobs, we were changing the location of the tile with PA and JS.

Results for the largest examined mesh/torus (size  $15 \times 10$ ) with VC flow control are presented in Tables I and II. Meshes and toruses with other considered sizes confirm the outcomes. The EVC flow control in all cases improves energy characteristic, as it is reported in [9], [21]. For  $15 \times 10$  CMP, we analyzed scenarios where a tile with PA and JS is in location:  $\langle 0, 0 \rangle$ ;  $\langle 0, 2 \rangle$ ;  $\langle 0, 5 \rangle$ ;  $\langle 0, 7 \rangle$ ;  $\langle 0, 9 \rangle$ ;  $\langle 3, 0 \rangle$ ;  $\langle 3, 2 \rangle$ ;  $\langle 3, 5 \rangle$ ;  $\langle 3, 7 \rangle$ ;  $\langle 3, 9 \rangle$ ;  $\langle 7, 0 \rangle$ ;  $\langle 7, 2 \rangle$ ;  $\langle 7, 5 \rangle$ ;  $\langle 7, 7 \rangle$ ;  $\langle 7, 9 \rangle$ ;  $\langle 10, 0 \rangle$ ;  $\langle 10, 2 \rangle$ ;  $\langle 10, 5 \rangle$ ;  $\langle 10, 7 \rangle$ ;  $\langle 10, 9 \rangle$ ;  $\langle 14, 0 \rangle$ ;  $\langle 14, 2 \rangle$ ;  $\langle 14, 5 \rangle$ ;  $\langle 14, 7 \rangle$ ; and  $\langle 14, 9 \rangle$ . Table I and Fig. 5 contain NoC energy consumption during simulation.

As we can see, in both 2D-Mesh (Tab. Ia) and 2D-Torus (Tab. Ib) CMPs, the lowest consumption of NoC energy is reported, when PA and JS are located in tile  $\langle 7, 5 \rangle$ . A general observed trend is that the locations in the middle of the CMP deliver higher energy savings in comparison to locations on

TABLE I  
NoC ENERGY CONSUMPTION IN [ $\mu$ J], BASED ON LOCATION OF PA AND JS: A) 2D-MESH, B) 2D-TORUS

9	21.19	17.48	15.68	16.94	21.94
7	18.70	15.00	13.33	14.57	19.72
5	17.68	14.08	12.35	13.59	18.52
2	19.03	15.34	13.59	14.84	20.07
0	21.56	17.84	16.30	17.58	22.25
	0	3	7	10	14

(a)

9	16.24	16.00	15.82	16.06	16.07
7	15.07	14.92	14.80	14.97	15.08
5	14.02	13.94	13.79	13.96	14.27
2	15.41	15.33	14.97	15.16	15.41
0	16.18	16.22	16.04	16.27	16.23
	0	3	7	10	14

(b)

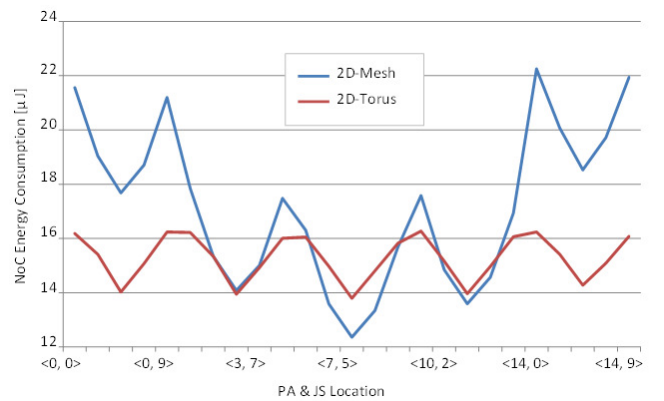


Fig. 5. NoC energy consumption in [ $\mu$ J], based on location of PA and JS.

the edges. The worst situation is noticed when corners are used as the location. The saving is especially visible in 2D-Mesh CMP, where by locating the PA and JS in the middle of the grid, we can save up to 45% of NoC energy. In 2D-Torus case, the saving is up to 16%. In the torus case, the saving is lower due to wrap-around channels that ensure shorter paths among tiles regardless the path traversed by a packet. Thus, the torus CMP is NoC energy saver and the saving does not depend so strongly on PA and JS location. The NoC energy results also confirm the conclusions reported in [9], that in general, 2D-Torus topology delivers better load balance and energy characteristic.

Energy used by PA and JS during simulation, in terms of their tile location on the chip, is shown in Table II and Fig. 6.

Table IIa contains results for 2D-Mesh while 2D-Torus is covered in Table IIb. In both cases, the energy saving gained due to adjusting PA and JS location is up to 1% only so, it can be neglected. Slight differences among reported results are caused by characteristics of allocation algorithm used. PA with BMAT algorithm for 2D-Torus consumes almost 6 times more energy than IFF technique for 2D-Mesh. The characteristic was also observed and discussed in [9].



TABLE II  
PA AND JS ENERGY CONSUMPTION IN [ $\mu$ J], BASED ON THEIR LOCATION  
IN CMP: A) 2D-MESH, B) 2D-TORUS

9	765.63	762.92	763.18	765.24	762.41
7	763.82	765.11	762.79	762.02	762.67
5	763.31	763.44	763.70	762.54	764.85
2	763.82	762.92	763.82	763.44	763.18
0	764.60	761.51	763.44	763.82	762.41
	0	3	7	10	14

(a)

9	4555.00	4542.73	4545.78	4560.37	4547.33
7	4550.40	4550.40	4552.70	4547.33	4548.10
5	4546.56	4547.33	4548.86	4541.96	4555.77
2	4548.86	4541.96	4549.63	4546.56	4556.53
0	4542.73	4548.86	4548.86	4551.16	4541.20
	0	3	7	10	14

(b)

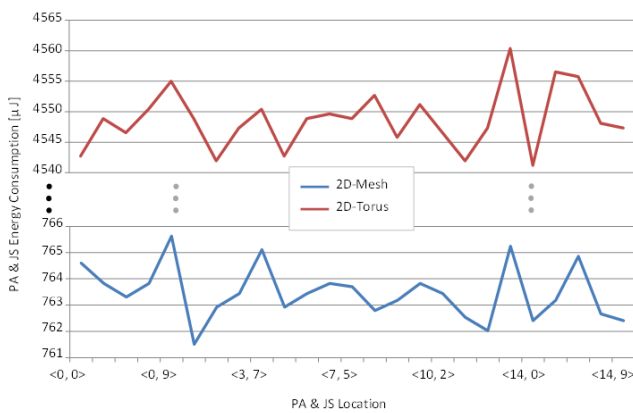


Fig. 6. PA and JS energy consumption in [ $\mu$ J], based on their location in CMP.

During simulation, computation energy was also calculated. It is energy consumed by all PEs to process all jobs in a queue. Each job in a queue has an associated number of cycles required – the number was randomly generated using discrete uniform distribution. For  $15 \times 10$  CMP, energy needed to process all jobs is 348.53 [mJ].

Considering all components impacting energy consumption,  $15 \times 10$  CMP with 2D-Mesh NoC consumes 349.31 [mJ] of energy during entire simulation. In 2D-Torus case it is 353.09 [mJ].

#### IV. CONCLUSIONS

In this paper, we have researched the impact of locating the tile with PA and JS on performance and energy consumption in CMPs. We have provided the mathematical energy models used in the research, and physical and logic structures of the system. The presented CMP model was simulated using an experimentation system.

The simulation results confirm that in general, 2D-torus CMP delivers better load balance and energy characteristic in comparison to 2D-Mesh structure. However, location of tile with PA and JS on the chip has significant impact on NoC

energy. By placing the tile in the middle of a chip, we can reduce the energy consumption by up to 16% (2D-Torus) and 45% (2D-Mesh). Moreover, by carefully placing the tile in a 2D-Mesh CMP, we can reduce NoC energy consumption by 11% in comparison to 2D-Torus. Thus, the 2D-Mesh driven by DOR-LB routing algorithm, delivers the best processor allocation solution for modern CMPs. The worst NoC energy characteristic is observed, when the tile is placed in the corners of a chip. It has been shown as well that location of the tile has minor impact on energy consumed by PA and JS in both the mesh and torus cases (we can save up to 1% only).

Finally, considering the total energy consumption (NoC, PA and JS, and PEs), 2D-Mesh CMP with PA and JS located in the middle of the chip delivers the best energy and load balance characteristics.

#### REFERENCES

- [1] N. Satish, C. Kim, J. Chhugani, A. D. Nguyen, V. W. Lee, D. Kim, and P. Dubey, "Fast sort on cpus, gpus and intel mic architectures," Intel, Tech. Rep., 2010.
- [2] D. Zydek, N. Shlayan, E. Regentova, and H. Selvaraj, "Review of packet switching technologies for future NoC," in *Proceedings 19th International Conference on Systems Engineering (ICSEng 2008)*, 2008, pp. 306–311, DOI: 10.1109/ICSEng.2008.47.
- [3] E. Salminen, A. Kulmala, and T. D. Hamalainen, "Survey of network-on-chip proposals," in *White Paper, OCP-IP*, 2008, pp. 1–13.
- [4] S. Uhrig, B. Shehan, R. Jahr, and T. Ungerer, "A two-dimensional superscalar processor architecture," in *Proceedings of the 2009 Computation World (COMPUTATIONWORLD '09)*, 2009, pp. 608–611, DOI: 10.1109/ComputationWorld.2009.46.
- [5] W. J. Dally, "Performance analysis of k-ary n-cube interconnection networks," *IEEE Transaction on Computers*, vol. 39, no. 6, pp. 775–785, 1990, DOI: 10.1109/12.53599.
- [6] D. N. Jayasimha, B. Zafar, and Y. Hoskote, "On-chip interconnection networks: Why they are different and how to compare them," Intel, Tech. Rep., 2006.
- [7] D. Zydek and H. Selvaraj, "Hardware implementation of processor allocation schemes for mesh-based chip multiprocessors," *Journal of Microprocessors and Microsystems*, vol. 34, no. 1, pp. 39–48, 2010, DOI: 10.1016/j.micpro.2009.11.003.
- [8] —, "Fast and efficient processor allocation algorithm for torus-based chip multiprocessors," *Journal of Computers & Electrical Engineering*, vol. 37, no. 1, pp. 91–105, 2011, DOI: 10.1016/j.compeleceng.2010.10.001.
- [9] D. Zydek, H. Selvaraj, L. Koszalka, and I. Pozniak-Koszalka, "Evaluation scheme for noc-based cmp with integrated processor management system," *International Journal of Electronics and Telecommunications*, vol. 56, no. 2, pp. 157–168, 2010, DOI: 10.2478/v10177-010-0021-4.
- [10] W. J. Dally and B. Towles, *Principles and Practices of Interconnection Networks*. San Francisco: Morgan Kaufmann, 2004.
- [11] G. Michelogiannakis, D. Sanchez, W. J. Dally, and C. Kozyrakis, "Evaluating bufferless flow control for on-chip networks," in *Fourth ACM/IEEE International Symposium on Networks-on-Chip (NOCS '10)*, 2010, pp. 9–16, DOI: 10.1109/NOCS.2010.10.
- [12] T. Moscibroda and O. Mutlu, "A case for bufferless routing in on-chip networks," *ACM SIGARCH Computer Architecture News*, vol. 37, no. 3, pp. 196–207, 2009, DOI: 10.1145/1555815.1555781.
- [13] L. B. Daoud, M. E. Ragab, and V. Goulart, "Faster processor allocation algorithms for mesh-connected cmps," in *Proceedings of 14th Euromicro Conference on Digital System Design (DSD 2011)*, 2011, pp. 805–808, DOI: 10.1109/DSD.2011.107.
- [14] B. S. Yoo and C. R. Das, "A fast and efficient processor allocation scheme for mesh-connected multicomputers," *IEEE Transaction on Computers*, vol. 51, no. 1, pp. 46–60, 2002, DOI: 10.1109/12.980016.
- [15] D. Zydek, H. Selvaraj, G. Borowik, and T. Luba, "Energy characteristic of processor allocator and network-on-chip," *Journal of Applied Mathematics and Computer Science*, vol. 21, no. 2, pp. 385–399, 2011, DOI: 10.2478/v10006-011-0029-7.
- [16] Y. T. Chan, Y. Z. Elhalwagy, and S. M. Thomas, "Estimation of circle parameters by centroiding," *Journal of Optimization Theory Applications*, vol. 114, no. 2, pp. 363–371, 2002, DOI: 10.1023/A:1016087702231.

- [17] A. Shawky, A. Ordys, and M. J. Grimble, "End-point control of a flexible-link manipulator using state-dependent riccati equation technique," *Archives of Control Sciences (ACS)*, vol. 12, no. 3, pp. 191–207, 2002, DOI: 10.1109/CCA.2002.1040236.
- [18] A. Shawky, A. Ordys, L. Petropoulakis, and M. Grimble, "Position control of a flexible-link manipulator using nonlinear h with state-dependent riccati equation," *Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering*, vol. 221, no. 3, pp. 475–486, 2007, DOI: 10.1243/09596518JSCE313.
- [19] D. Zydek, H. Selvaraj, and L. Gewali, "Synthesis of processor allocator for torus-based chip multiprocessors," in *Proceedings of 7th International Conference on Information Technology: New Generations (ITNG 2010)*, 2010, pp. 13–18, DOI: 10.1109/ITNG.2010.145.
- [20] Intel. (2011, Sep) Intel microprocessor export compliance metrics. [Online]. Available: <http://www.intel.com/>
- [21] A. Kumar, P. K. L. S. Peh, and N. K. Jha, "Express virtual channels: Towards the ideal interconnection fabric," *ACM SIGARCH Computer Architecture News*, vol. 35, no. 2, pp. 150–161, 2007, DOI: 10.1145/1273440.1250681.