International Journal of
Science and Engineering Investigations vol. 5, issue 53, June 2016
ISSN: 2251-8843

Improving the GPU Performance Prediction Models to Design Space Exploration

Morteza Sadeghi
Engineering and Computer Science Sector at the University of Shiraz
(mortezamsp.2014@gmail.com)

Abstract- Recently, GPUs have also been used plenty in the scientific calculations for high-performance in parallel computing power and low energy consumption. Offering GPU performance prediction models base on Micro-architecture parameters to optimal design in the hardware process of the GPU, has been the subject of prior works. In this article, we defined design space NVIDIA Fermi GPU bigger than the prior similar work and with a 264 minimum size point, and we made performance prediction models by sampling only 45,000 design points from that, and then we offer an efficient search algorithm that carefully explore the design space by helping the performance models. Finally, we compare the our models with models of previous work by the help of this algorithm, and find out, that ,making models by more parameters and levels of values, help to more carefully explore in design space. Also, by analysis the models and results, we analyze the program estimator behavior with respect to micro-architecture parameters.

Keywords- *Design Space Exploration, GPU, Micro-architecture, Performance prediction models*

I. INTRODUCTION

Design Space Exploration, is known to investigate the all possible designs and then choosing the best design selection among them. Design Space Exploration has been investigated for GPU in several works as yet. For example, one work [1] is finding the best Design of GPU by consider to limitation in sources, and another same work, offers the models that predict the behavior of GPUs based on the value of parameters in micro-architecture. Using these models, there is no need to implement different designs GPUs to choose the best case. By helping these models, we can design optimal graphics processor based on resource constraints. We improve the earlier works to make performance prediction model and show that to do more simulations, we will achieve the models with better performance. We improved the presented models of Mr. Mirsoleimani [3] that it can explore more carefully, and analyze the more design space. We make create more than 46,000 simulations performed in 2 months for making our models. By taking this time, we now have models which can predict 10 benchmark applications of implemental IPC on any processor NVIDIA Fermi. Innovations in this article can be summarized as follows;

- Improving the performance prediction model of NVIDIA Fermi GPUs
- Presenting the algorithm of efficient search to Design Space Exploration of graphics processor in relation to specific application.
- Analysis the behavior of benchmark applications with regard to value of micro-architecture parameters

II. RESEARCH BACKGROUND

Presentation the performance prediction models of GPUs has been the subject of the recent works. We have adopted our main idea of this paper from [1] and [2] and [3]. Jia in work [2] has presented micro-architecture of the performance prediction models of GPUs by considering 32 parameters and he showed that in order to make a highly accurate prediction model it is sufficient to study only 1% of total design space. In other work [1], Jooya showed that the number of the necessary simulations to identify the most influential parameters of micro-architecture of the performance can be reduced by using [6] design method. Then after, Mirsoleimani [3] proposed an efficient model by combining Jooya [1] and Jia [2] models to make the performance prediction models for design space explorations in GPUs which reduces the number of necessary simulations; then, according to this model the prediction models are proposed for the performance of GPUs of the NVIDIA Fermi architecture model. The presented models of Mirsoleimani [3] work have had prediction error less than 5%.

The presented prediction models in three works support the huge design space, including not made devices, as well. While [4] have proposed models which include small areas around made devices. The purpose of this work has been to reduce the search space and remove a large part of it.

One of the problems of the presented work has been the selection of the primary set of the influential parameters on performance criteria based on the researchers' defaults. Jia [2] made his model with just 32 micro-architecture parameters without mentioning the reason. Mirsoleimani [3] also did not mention the reason of the selection of his own primary set of 32 parameters, though he had used Plackett & Burman [6] model to reduce their numbers. While both of the works were done by using gpgpu-sim simulator software [8]; the software

can support 64 micro-architecture parameters. Another problem of the prior works is that in order to estimate the performance of a specified processor they can't be used because the low number and low level of values of parameters support low amount. We can only find the best processors of architecture of Fermi NVIDIA with the help of these models or we can use the proposed model to find the best processor in order to run a special program.

In the work [10], the optimal hardware design of the GPU has studied in terms of performance and energy. In this work lower than 10 micro-architecture parameters have studied and like [2] and [3]. The reason of selection these set of parameters has not been mentioned.

In the work [11] different methods have studied for modeling to be used in DSE and it proposed a method which have a higher prediction accuracy and needs less sampling than linear regression (that was used in [2] and [3]). Jia [2] also improved his prior method in [12] and he has used the tree of regression instead of linear regression. However, we use this method in our work due to the simplicity of the regression method and compare our work results with work [3]. In addition, according to [2] and [3] works the error of regression model to predict IPC of GPUs is acceptably less than 6%.

III. RESEARCH METHODOLOGY

Based on the problems of the prior works, it is first needed to put aside their defaults in the selection of the primary set of the influential parameters on performance criteria and use all possible parameters. It is also necessary to make better models with more parameters and higher level of values for each parameter in order to have a good performance prediction model to explore the design space. In this paper two categories of the new model were made each of which meets each of the above needs.

To this end, we have first made new models by considering larger design space than the works of [3] we use all the 64 parameters of micro-architecture supported by gpgpu-sim software [8]. After that we made other models once more, in which not only the design space is improved, but also the numbers of levels of values of its parameters are increased. Then we compare the exploratory power of the design space with the models and we showed that the new models have better performance to explore the design space. In this paper, we have used the terms “the first improved models” and “the second improved models” to refer to each of the new models. The design space is improved in the first improved models and the numbers of the parameters and their levels are increased in the second improved models. Finally, the behavior of the benchmark applications is also analyzed due to the changes of micro-architecture parameters numbers.

A. Increasing the number of the design space parameters

In the work of [3] there were 32 parameters of the design space which were considered on the basis of the researcher’s defaults. However, the simulation software, gpgpu-sim [8], used in their work can use up to 64 micro-architecture parameters to simulate the GPUs [8]. The parameters are shown in table 1 which is attached at the end of this paper. Therefore, we have put aside the defaults of the prior works [3] by adopting the dihedral (tow-levels) values for the parameters we have simulated 128 different states and in the phase of design of State-Space we consider 64 parameters instead of their desired 32, but the rest of the modeling work is done as such as they did. Thus, such as Plackett & Burman [6] for each of the 10 benchmark applications were used to select 7 parameters of the more important ones among 64 parameters. After that the regression models were made which calculated the value of log IPC of the benchmark applications given the important 7 parameters. It is worth mentioning that in the prior works [2] and [3] the Logarithm of the IPC was used to reduce the prediction error of the model and we also use the same parameter in our work. At least 320 simulations are required to make new models.

In table 1, fourth to eighth column in the table above shows the allowable values for each micro-architecture parameter. Performance Prediction models have been made based on these values. Also the number of allowed values of some parameters is 10 or more, that are not shown due to the length, in the table. These parameters are marked with an asterisk * in the table. Values of parameters 7 to 10, 10 is the value that its maximum and minimum values are only displayed in the table. Other values are in equal ranges between these two values, and Parameters 55, 56, 58 and 59 are like this. 12 parameter values are multiples of 16 between the 96 and 1632. 13 parameter values are even number between 4 and 32. Values Parameters of 25, 30, 35, 39, and 43 are calculated by the values of their next three parameters.

As it is shown in table 2, our models have had prediction error less than 3%. Error calculation is done according to equation (1) same as the work of [3]. We have studied the performance of the made models to our work’s models [3].

$$err = \frac{Predicted\_IPC\_\log - Simulated\_IPC\_log}{Simulated\_IPC\_log} \quad (1)$$

TABLE I. USED MICRO-ARCHITECTURE PARAMETERS IN THIS WORK

	Parameter's name	**unit**	**The allowable levels of parameters**				
1	n_clusters	num	4	8	10	15	20
2	topology_k	num	15				
3	n_cores	num	1	6	12		
4	dram_sub_partitions	num	2	4			
5	nbnk	num	12	16			
6	n_mem_ctrlr	num	3	6			
7	Core_Clock*	MHz	500	900			
8	Interconnect_Clock*	MHz	500	900			
9	L2_Clock*	MHz	500	900			
10	DRAM_Clock*	MHz	700	1100			
11	n_registers	Knum	8	16	32	64	128
12	n_threads*	num	96	1536			
13	simd_width*	num	4	32			
14	warp_parts	num	1	2	4	8	

	Parameter's name	unit	The allowable levels of parameters				
15	n_blocks	num	1	2	4	8	16
16	ID_OC_SP	num	2	3	4		
17	ID_OC_SFU	num	1	2	3		
18	ID_OC_MEM	num	1	2	3		
19	OC_EX_SP	num	2	3	4		
20	OC_EX_SFU	num	1	2	3		
21	OC_EX_MEM	num	1	2	3		
22	EX_WB	num	2	3	4		
23	sp_units	num	2	3	4		
24	sfu_units	num	1	2	3		
25	dl1_size*	KB	2	512			
26	dl1_nsets	num	16	32	64		
27	dl1_bsize	byte	128	256			
28	dl1_assoc	way	2	4	8		
29	shmem_size	Kbyte	16	32	48	64	128
30	dl2_size*	KB	4	1024			
31	dl2_nsets	num	32	64			
32	dl2_bsize	byte	128	256			
33	dl2_assoc	way	2	4	8		
34	dl2_text	mode	0	1			
35	il1_size*	KB	0	16			
36	il1_nsets	num	2	4			
37	il1_bsize	byte	128	256			
38	il1_assoc	way	2	4			
39	tl1_size*	KB	2	512			
40	tl1_nsets	num	4	8			
41	tl1_bsize	byte	128	256			
42	tl1_assoc	way	8	24	32		
43	cl1_size*	KB	2	256			
44	cl1_nsets	num	32	64	128		
45	cl1_bsize	byte	64	128			
46	cl1_assoc	way	2	4			
47	o_collect_sp	num	6	8	16		
48	o_collect_sfu	num	8	16	32		
49	o_collect_ip_sp	num	1	2	16		
50	o_collect_op_sp	num	1	2	16		
51	n_reg_banks	num	2	8	16		
52	shmem_nbanks	KB	8	16	32	64	128
53	shm_broad_cast	num	0				
54	max_issue	num	1	2	4		
55	dl2_lat*	cycle	120	130			
56	dram_lat*	cycle	100	110			
57	dram_shceduler	mode	0	1			
58	dram_sched_qsize*	num	1	16			
59	Dram_rqsize*	num	116	132			
60	n_chips	num	1	2	8		
61	dram_buswidth	bytes	2	4	8	16	
64	dram_burstlength	num	4	8	16		
64	RTPL	num	2	8			
64	n_sched	num	2	4	8		

TABLE II. PREDICTION ERROR OF FIRST IMPROVED MODELS

	Prediction error (percent)	
benchmark applications	***Average***	***Maximum***
BFS	1.41	5.415
CP	0.501	2.596
LPS	0.483	7.163
NQU	0.001	0.003
RAY	0.028	0.116
STO	0.089	0.283
BP	1.479	7.182
BFSr	0.399	2.901
HS	0.272	1.74
NW	2.319	8.072

B. Increasing the number and levels of the model parameter values

We made other models in order to more exact study the design space of GTX480 in which both the levels of parameter values and the number of parameters in the model, from 7 to 20, be increased. The Plackett & Burman [6] design cannot be used any more due to the increase of parameter levels because this method is used for dihedral (two-level) parameters; therefore, a Factorial Fractional Design is used [7]. First, 1500 design points are randomly selected from the design space. Each of the design point is a micro-architecture's configuration and in fact shows a specific GPU. The resolution of the configurations is increased and 5 parameters are changed in each configuration than to the prior one in order the relations between the parameters can be considered by fractional factorial design. Then, the performance of all 10 benchmark applications on all 1500 different designs were simulated by using the simulator software and the IPC values were calculated from the simulator's output for all of the estimators. Then, 20 parameters were selected with the most influence on IPC values were selected with Feature Selection, the automatic tool of the Statistica software [5] that acts on the basis of ANOVA; the design space was redefined based on these 20 parameters. The software has also been used to make the regression model in the works [2] and [3]; therefore, the design space became smaller by ignoring the parameters which had lee influence on IPC values In the next phase the regression model is made which the IPC value of each of the estimators' performance on every GPUs is predicted according to the values of micro-architecture parameters of the GPU. The state space for 20 parameters with different levels of values has at least 2020 design point. Therefore, all of the design points cannot be simulated to make the regression model because it is time-consuming to simulate all of them. Then only 1500 random points are selected among 2020 points for each of the estimators and the IPC of every 10 estimators' performance were calculated by the simulation. Then the regression model was made for each estimator and it was used to predict the rest of the design space points. For the third time we have randomly selected 1500 points of design space and then after they were simulated they were used to test our mode's errors; therefore 4500 simulations were done to make the second improved models.

Due to the large number of needed stimulations for each prediction models as well as due to long term implantation of BFSr and NW benchmark applications, We used a smaller size data than the work [3] to run this program, that is why the two models mentioned above, was not comparable with the work of [3]. Created models have the average prediction error less than 7%. In Table 3, prediction error model is shown for each model.

TABLE III. PREDICTION ERROR OF SECOND IMPROVED MODELS

	Prediction error (percent)	
benchmark applications	***Average***	***Maximum***
BFS	2.79	32.94
CP	0.3	0.18
LPS	4.69	32.93
NQU	0.67	6.44
RAY	0.03	0.25
STO	3.28	32.18
BP	2.87	22.84
BFSr	6.31	33.1
HS	4.24	24.56
NW	0.99	7.39

IV. RESEARCH FINDINGS

In this section, first, Created models are compared with each other, and then analyzed the behavior of benchmark applications according to regression models and parameters of micro- architecture.

A. Comparing Models

New models work and previous work [3] have been made with good accuracy. Accuracy of first improved models is less than 5%, and second improved models are less than 7%. The two tests have been conducted in order to comparisons the power of models exploration. Once, we examined the work models [3] with first improved models, and once we examined the first improved models with second improved models in order to power of space exploration. In each test, superiority criterion of two prediction models, finding machines with more IPC in the area of design that have been covered by predictive models. Space that is supported by previous models [3] and first improved models is maximum $2^7 = 128$ state. Therefore, finding the best design that is supported by these models is obtained by comparing the 128 mode. But for second improved models, supported space is near $6*10^{128}$ states. So a single-objective algorithm should be used to find the best design. To solve this problem, we offer a genetic algorithm and improved its parameters to be able to explore search space more carefully. Explanations that are related to genetic algorithm, were removed due to the increasing the volume of paper.

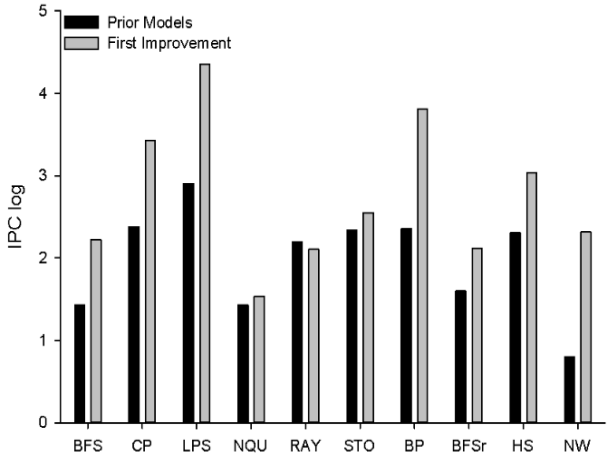

Figure 1. Comparing the logarithm of IPC, best found device for work models [3] and the first improved models.

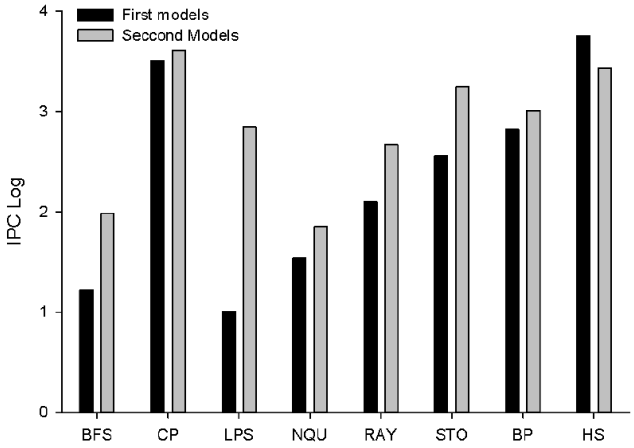

Figure 2. Comparing the logarithm of IPC, best found device by first improved models, and second improved models.

1) The impact of improved design of the space - state in the design space exploration

In this section, work models [3] are compared with first improved models. Each models work [3] and the first improved models can only predict 128 points in the design space. For each of the models and benchmark applications, found best design point (design with maximum IPC) out of 128 points. Then, for each benchmark applications, we compared the best place to found design model by work model [3] and first improved models. As Chart 1, the first improved model in which 32 parameters were used more than the work [3] for its space design, indicates better results, and it indicates assumptions of work [3] were incorrect and 32 parameters that were ignored, can help to the better work.

2) The impact of increasing the number of parameters and their values on the design space exploration

It is predictable that the increase in the number and levels of parameter values will lead to a more accurate model. In the second improved models, in addition to the number of parameters, values are also increased. In the diagram (2) the best results of the second improved models are compared with the first improved models. In this comparison, both models are compared with each other in a similar search space, so that the minimum and maximum values of each parameter are the same for both models, but the numbers of values are different. Diagram (2) shows that in the same range of parameter values, increasing the number of parameter values leading to better results (except in one benchmark). In this comparison, assessment programs of BFSr and NW have been removed from comparisons because of the difference of their input data. This can be explained on the HS evaluation that the increase of the levels made the search space more complex and thus probably search algorithm failed to find better answer than the first improved models.

B. Behavior analysis of benchmark applications according to the parameters of micro-architecture

One of the interesting results that were obtained during this study was that increasing the speed of benchmark applications were not identical by consider to improving the hardware parameters. By increasing the amount of micro-architectural parameters, the speeds of some programs have hardly increased. However, the speed of some of benchmark applications has been increased.

To illustrate this issue in second improved models, we take attention to 32 new parameters compared to work [3] with a constant value and equal to its corresponding value of those parameters in hardware GTX480. And then by using a genetic search algorithm presented in this work, we searched for the best device for any benchmark applications. The comparison is shown in the figure 3. This figure also shows the logarithm value of the IPC running the programs on the device's GTX480.

As figure 3, best found designs for some of the benchmark applications despite the large search space, there is not increasing in speed, compared to the limited search space. For example, benchmark applications NW have been improved very much due to the higher sensitivity compared to parameter memory latency promoter dl2_lat, and increasing the value of this parameter in the search space new.

In contrast, the best found design for the programs such as NQU didn't have much speed increasing compared to best found design in limited search space. NQU program more than any other parameters, are depended on Wrap Parts parameters and its increasing speed has reversed state with increasing the amount in Warp_Parts. Considering that, value of this parameter cannot be reduced less than 1 and the value of this parameter is 2in the GTX480, so this program cannot be increased more than. We conclude that the program NQU cannot be run on each GPU model of Fermi, more than 2 IPC_Log value, and this limitations is related to program NQU.

In the benchmark applications BP, STO, CP, LPS, HS, the "n_cores" parameter that has been ignored in prior works, has a large influence on the speed increasing. This parameter increases the power of parallel run programs. The presence of this parameter in the search space, cause in finding new architectures that increase the run speed of mentioned benchmark applications.

C. Studding the benchmark applications regarding to the regression model

To making regression models for Second improved models, 20 parameters are used for each benchmark applications with the greatest impact in the IPC. Comparing the values of the parameters in the regression model can provide more information about the benchmark applications.

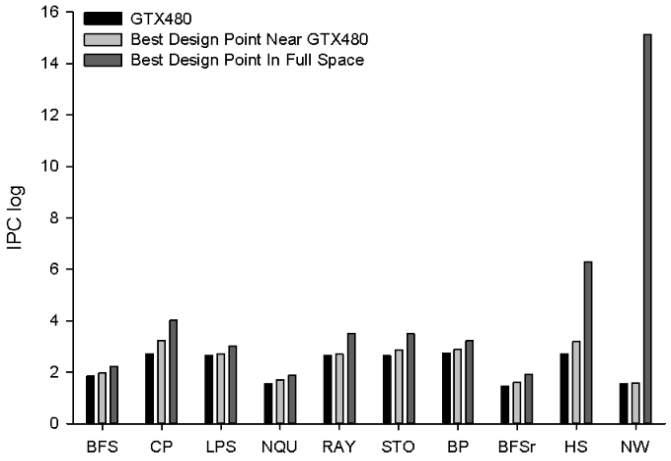

Figure 3. Comparing the logarithm of IPC, best found device by second improved models and limited search space, second improved models in full search space, and logarithm of IPC of GTX480**.**

In figure (4) to (13), some parameters are shown with the highest ratio in regression model of each of. As seen in these figures, parameters (n_threads, n_cores, simd_width, wrap) exist in most benchmark applications as an important parameter in the regression model. These parameters have the most influence on the IPC.

So by increasing these, all configurations of the IPC will be increased. Consider the benchmark applications, in this function program, SQRT have been used in graphics card code more. This function is created in the special functions (SLI) graphics card, and both simulation software of GPGPU-Sim [8] and also parameter (num_SFU_units) are associated with it. That's why this is the third important effective parameter on IPC. This is one of the parameters that was omitted in the prior works [3]. Also, a subtraction and multiplication function in the code of the graphic card is used to determine the correct location of ministers in benchmark applications. These operations are run in graphics card at section of ALU that associated hardware parameter is in GPGPU-Sim parameter of

num_SP_units. So, as CP parameter num_SFU_units, is one of the most influential parameters on the IPC's in this program.

This case is shown in Figure (7). This is true on the program STO, because in this the program, also much energy functions is used in the code of the graphics card, that the function is run in the graphics card in the special function section. Figure 9, shows the important STO parameters.

The programs BFS version of ISPASS [8] and version RODINIA [9] (here named BFSr), BFS search algorithm is used to graph. With respect to running these programs, the size of the general memory is the most influential important parameters of hardware PGU on IPC. According to the figure (4) and (11), it is shown that parameters of the memory have an impact on the value of IPC.

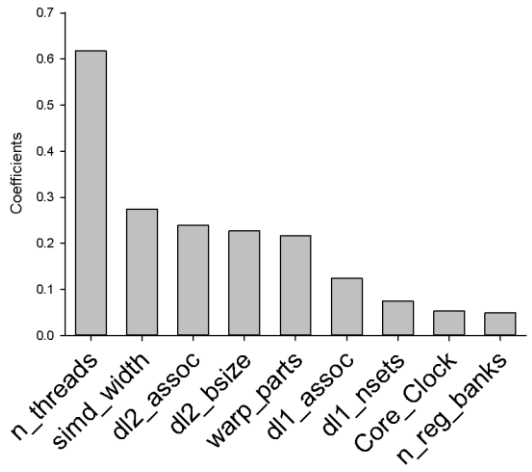

Figure 4. The most important coefficients of BFS model

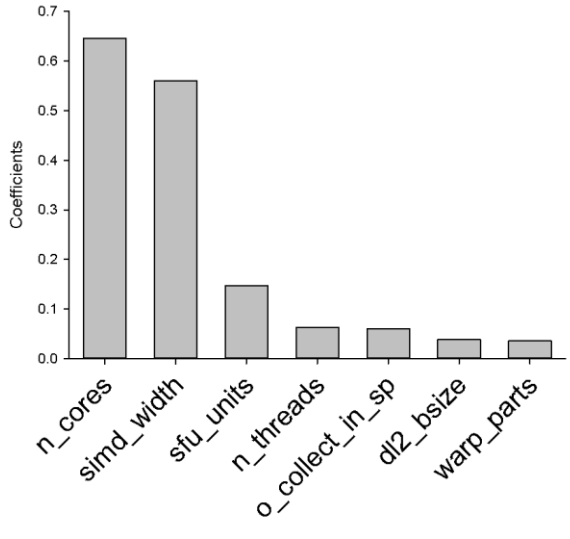

Figure 5. The most important coefficients of CP model

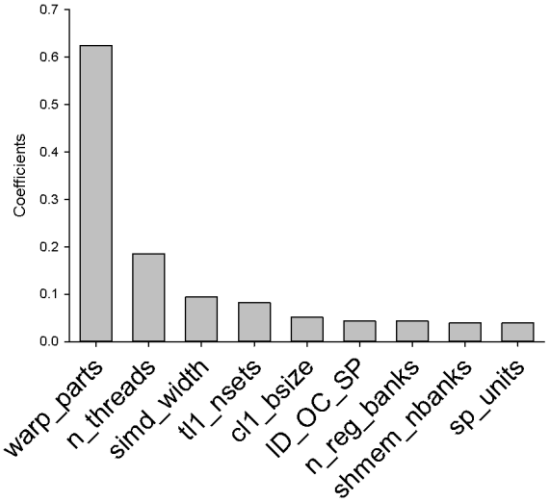

Figure 6. The most important coefficients of LPS model

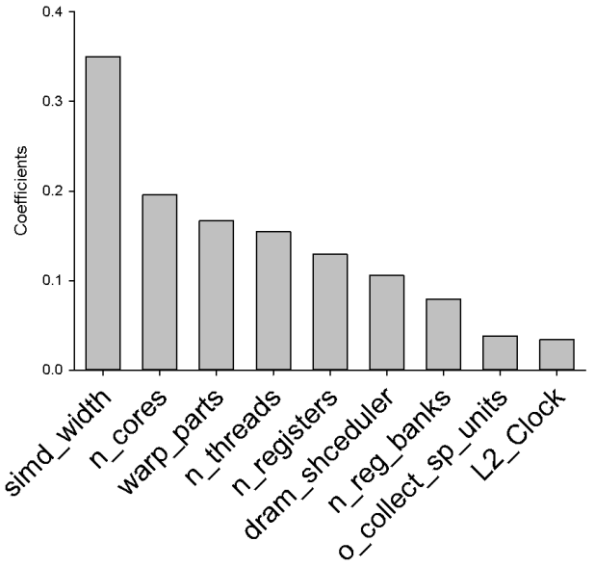

Figure 7. The most important coefficients of NQU model

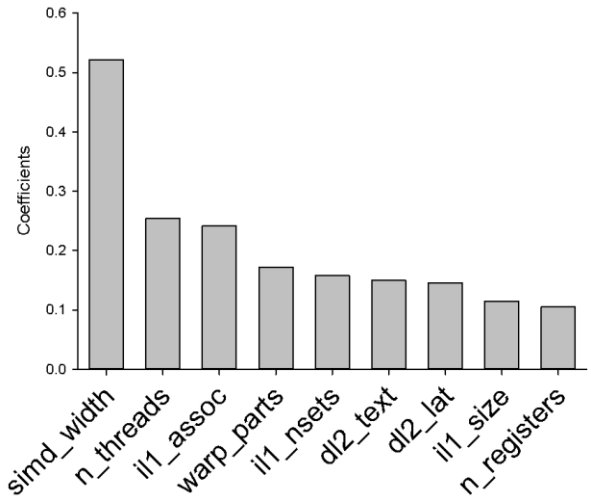

Figure 8. The most important coefficients of RAY model

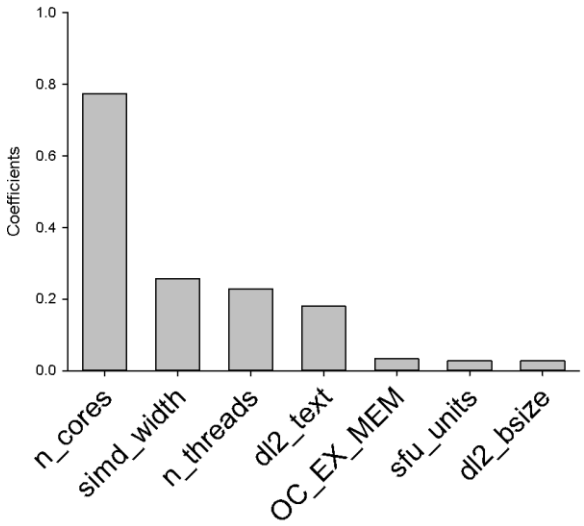

Figure 9. The most important coefficients of STO model

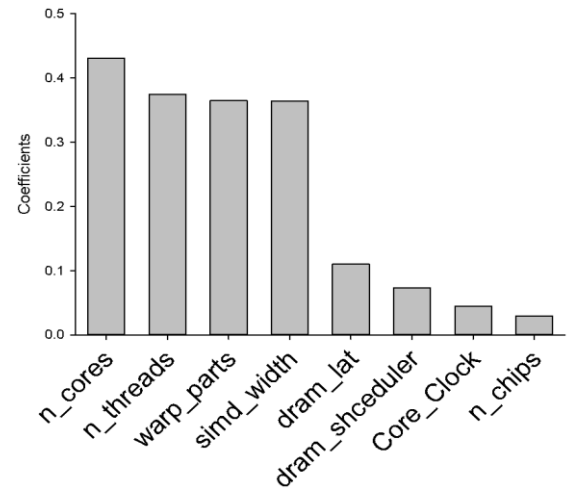

Figure 10. The most important coefficients of BP model

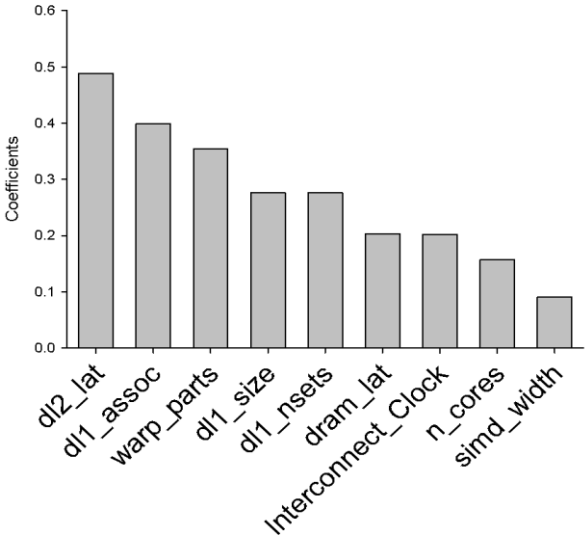

Figure 11. The most important coefficients of BFSr model

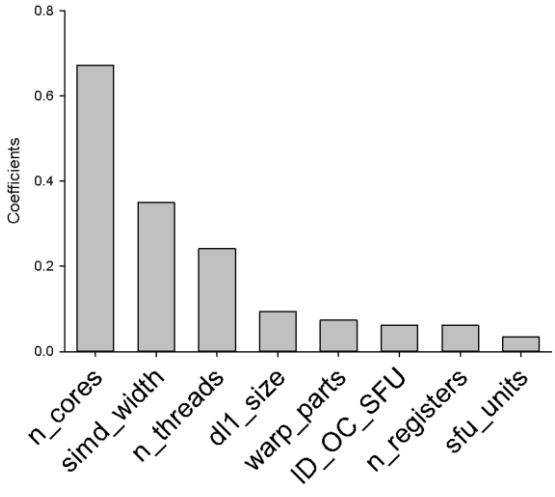

Figure 12. The most important coefficients of HS model

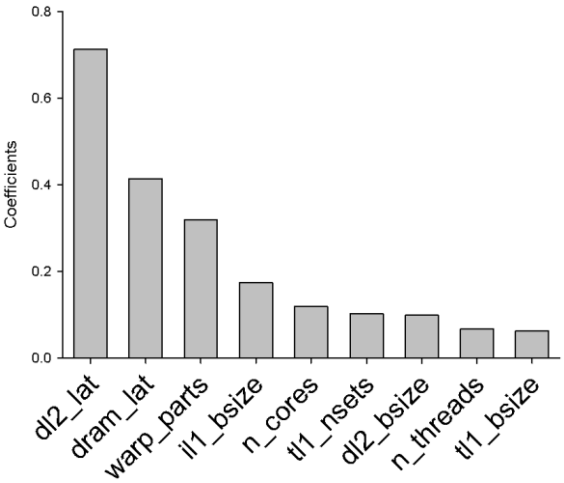

Figure 13. The most important coefficients of NW model

Other memory-related programs such as CP, EQU, STO that is used of the loading and storage operations due to the high interaction between thread, are highly related as much as block_size and Caches. NW and BFSr programs are highly dependent on memory bandwidth, for this reason, parameters dram_lat and dl2_lat is one of the influential important parameters on the value of IPC in these programs.

V. CONCOLUSION

The results of tests conducted in this paper, to explore the detailed design space exploration of a GPU, if we increase the values of parameters and value levels in the design space state, we will be able to achieve the better results. Also we found that prior knowledge of researcher about the impact of micro-architecture parameters on processor performance is not reliable and it should be a test for the initial set selection of parameters.

Searching for single-objective, it was observed that for some programs, if we don't improved the architecture , we cannot expect a large increase run speed for them. These benchmark applications should improve the factors such as program code to further run speed. It is notable that the methods and results presented in this article can also be used for the other criteria of performance or GPU architectures. In addition, second improved models, can also be used for other devices except GTX480, due to their supported space.

REFERENCES

[1] Jooya, A., Baniasadi, A., & Dimopoulos, N. J. (2013, January). Efficient design space exploration of GPGPU architectures. In Euro-Par 2012: Parallel Processing Workshops (pp. 518-527). Springer Berlin Heidelberg.

[2] Jia, W., Shaw, K. A., & Martonosi, M. (2013, October). Starchart: Hardware and software optimization using recursive partitioning regression trees. In Proceedings of the 22nd international conference on Parallel architectures and compilation techniques (pp. 257-268). IEEE Press.

[3] Mirsoleimani, S. A., Karami, A., & Khunjush, F. (2014). A Two-Tier Design Space Exploration Algorithm to Construct a GPU Performance Predictor. In Architecture of Computing Systems–ARCS 2014 (pp. 135-146). Springer International Publishing.

[4] Zhang, Y., Hu, Y., Li, B., & Peng, L. (2011, July). Performance and power analysis of ATI GPU: A statistical approach. In Networking, Architecture and Storage (NAS), 2011 6th IEEE International Conference on (pp. 149-158). IEEE.

[5] StatSoft, I. N. C. (2001). STATISTICA (data analysis software system), version 6. Tulsa, USA, 150.

[6] Plackett, R. L., & Burman, J. P. (1946). The design of optimum multifactorial experiments. Biometrika, 305-325.

[7] Montgomery, D. C., & Runger, G. C. (2010). Applied statistics and probability for engineers. John Wiley & Sons.

[8] Fung, W. W., Sham, I., Yuan, G., & Aamodt, T. M. (2007, December). Dynamic warp formation and scheduling for efficient GPU control flow. In Proceedings of the 40th Annual IEEE/ACM International Symposium on Microarchitecture (pp. 407-420). IEEE Computer Society.

[9] Che, S., Boyer, M., Meng, J., Tarjan, D., Sheaffer, J. W., Lee, S. H., & Skadron, K. (2009, October). Rodinia: A benchmark suite for heterogeneous computing. In Workload Characterization, 2009. IISWC 2009. IEEE International Symposium on (pp. 44-54). IEEE.

[10] Jarvis, S., Wright, S., & Hammond, S. D. (Eds.). (2014). High Performance Computing Systems. Performance Modeling, Benchmarking and Simulation: 4th International Workshop, PMBS 2013, Denver, CO, USA, November 18, 2013. Revised Selected Papers (Vol. 8551). Springer.

[11] Wang, H., Zhu, Z., Shi, J., & Su, Y. (2015, January). An accurate ACOSSO metamodeling technique for processor architecture design space exploration. In Design Automation Conference (ASP-DAC), 2015 20th Asia and South Pacific (pp. 689-694). IEEE.

[12] Jia, W., Shaw, K. A., & Martonosi, M. (2013, October). Starchart: Hardware and software optimization using recursive partitioning regression trees. In Proceedings of the 22nd international conference on Parallel architectures and compilation techniques (pp. 257-268). IEEE Press.

[13] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955. (references)

[14] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

[15] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in Magnetism, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.

[16] K. Elissa, "Title of paper if known," unpublished.

[17] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.

[18] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Japan, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[19] M. Young, The Technical Writer's Handbook. Mill Valley, CA: University Science, 1989.