Power-aware Performance of Mixed Precision Linear Solvers for FPGAs and GPGPUs

JunKyu Lee[1], Junqing Sun[2], Gregory D. Peterson[1], Robert J. Harrison[3], Robert J. Hinde[3]

Electrical Engineering and Computer Science[1], Chemistry[3], University of Tennessee
Knoxville, Tennessee, USA
[jlee57, gdp, robert.harrison, rhinde]@utk.edu

Marvell Semiconductor Inc. [2]
Santa Clara, California, USA
[philsun]@marvell.com

***Abstract*— Power has emerged as a significant constraint to high performance systems. We propose modeling power-based performance (performance/watt) and clock-based performance for GPGPUs and FPGAs. Based on the modeling, we perform a case-study with mixed precision linear solvers for a Xilinx XC5VLX330T FPGA and NVIDIA Tesla C1060 GPU. In the case-study, the FPGA shows power- and clock-based performance better than the GPGPU while the GPGPU shows better time-based performance.**

Keywords- Mixed precision; Power Performance; Comparison;

I. INTRODUCTION

GPGPUs and FPGAs seem to be competing with each other for high performance computing. Many computational science applications are computationally intensive causing huge power consumption. In high performance computing, it is a challenge to keep the power budget low while keeping high performance.

There have been many research efforts for performance modeling [1-2], [12-13]. Along with time-based performance, power-aware performance becomes a significant performance metric in high performance computing applications [14-15].

In this paper, we propose a power-aware performance modeling approach relating clock-based performance for GPGPUs and FPGAs. The power-aware performance explores the achievable number of Flops if one additional watt is provided.

We did a case-study with mixed precision linear system solvers for Xilinx XC5VLX330T FPGAs and NVIDIA Tesla C1060 GPUs with this modeling. Interestingly, FPGAs can employ arbitrary precisions for mixed precision solvers while GPGPUs can employ only single and double precision [5-7].

II. POWER-BASED PERFORMANCE

The total power consumption is described below [3]:

$$U = S + D = S + C \times V^2 \times f \quad (1)$$
$$U = S + (C \times V^2) \times f = S + \alpha \times f, \quad \alpha = (U - S)/f = D/f$$

where U is total power consumption, S is static power consumption, and D is dynamic power consumption. C is an effective capacitance, V is an operation voltage, and f is an applied clock rate. We assume:

$$0 \leq D \leq (U_{MAX} - S) \quad (2)$$

Since S generally is not related to the computation, we consider D for the modeling. Now, we define three different performance metrics as follows.

$$F \coloneqq \text{\# of Flops},$$
$$F_{CLK} \coloneqq \text{\# of Flops/clock-cycle} = F/f$$
$$F_{WATT\text{-}D} \coloneqq \text{\# of Flops/Watt} = F/D \quad (3)$$

One can consider total power for the system, but we are focusing on the incremental performance benefit of one additional watt. Hence, we consider only the dynamic power. The power-based performance modeling is as follows:

$$MAX(F_{WATT\text{-}D}) = F/(U_{MAX} - S) = F/(\alpha_{MAX} \times f) = F_{CLK}/\alpha_{MAX} \quad (4)$$

A user can gain insight into the power efficiency and seek to maximize the achievable Flops/Watt. U_{MAX} and S are found with ease from the specification sheet for accelerators (e.g. GPGPU) or using some tool (e.g. Xilinx Power Estimator). Therefore, we can simply find α_{MAX} by dividing $(U_{MAX} - S)$ by the corresponding clock rate. Based on (4), it is important for a designer to implement applications to achieve high-performance per clock cycle in order to save power.

III. CASE STUDY - MIXED PRECISION LINEAR SOLVERS

Mixed precision linear solvers employ more than one precision computation [4]. They employ lower precision for matrix decomposition ($O(n^3)$) to approximate the solution and higher precision to refine the solution ($O(n^2)$). Higher performance can be achieved without losing the higher precision accuracy for the solution. The idea and the algorithm were clarified in [4, 6]. In the case-study, mixed precision linear solver performance (time, clock, and power-based) is specified and compared for Xilinx XC5VLX330T and NVIDIA Tesla C1060 platforms.

A. Precision Decision for LU decomposition

To determine the appropriate precision according to condition numbers for LU decomposition with partial pivoting for mixed precision solvers, we make two assumptions based on the literature [4, 8]:

1. Computation time is dominantly governed by the matrix decomposition. Hence, we consider the LU decomposition performance as the mixed precision solver performance.

2. Mixed precision solver success primarily depends on a matrix condition number. Prior work suggests that a mixed precision solver is able to succeed when the condition number is less than the reciprocal of the working precision for LU decomposition. Therefore, we consider that as the success condition for the mixed precision solver.

Based on these assumptions, the precision is:

$$\text{Mantissa bit width (M)} = (\log_2(\text{condition number})) - 1$$
$$\text{Exponent bit width (E)} = \begin{cases} 8 & (\text{if } M \le 23), \\ 11 & (\text{if } 24 \le M \le 52) \end{cases} \quad (5)$$

According to (5), a GPGPU can employ single precision when $M \le 23$ or double precision when $24 \le M \le 52$ for LU decomposition while the FPGA can employ arbitrary precisions.

B. Perfomance Measurement

We employ MAGMA v0.2 [9] for a hybrid system (NVIDIA Tesla C1060 GPU and Intel Xeon 2.93 GHz) and [6] for the Xilinx XC5VLX330T FPGA. We measure the performance directly for the GPGPU and estimate the performance for the FPGA based on Xilinx ISE 11.4 Place and Route (PAR) and performance modeling using the equation (6). Based on the PAR, we apply 120 MHz for the performance modeling. Next, we seek the required number of DSP48Es according to various precisions. Table I shows the number of Processing Elements (PEs) that can fit on a single FPGA according to precision.

Next, we seek the number of PEs according to condition numbers based on the total number of DSP48Es for Xilinx XC5VLX300T. Table II shows the number of PEs based on (5) and Table I.

Table I. Number of PEs on Xilinx XC5VLX330T

Exponent	Mantissa	# of required DSP48Es per PE	Achievable # of PEs
8	16	1	192
8 (Single)	17-23	2	96
11	24-33	4	48
11	34-40	6	32
11	41-50	9	21
11	51	12	16
11 (Double)	52	10	19

Table II. Number of PEs on Xilinx XC5VLX330T

Condition Number	$1–2^{17}$	2^{17}-2^{24}	$2^{24}–2^{34}$	$2^{34}–2^{41}$	$2^{41}–2^{51}$	$2^{51}–2^{52}$	$2^{52}–2^{53}$
# of bits S+E+M	10-25	26-32	36-45	46-52	53-62	63	64
# of M	1–16	17-23	24-33	34-40	41-50	51	52
# of PEs	192	96	48	32	21	16	19

Each PE in the design of [6] has one multiplier and one adder for the computation. The performance on Xilinx XC5VLX330T is estimated as follows.

Performance (GFlops) on XC5VLX330T
= 2(Flops) × Number of PEs × Clock Rate (0.12 GHz) (6)

C. Power-based Perfomance

The static power consumption is 57.7 W, the peak is 187.8 W, and the clock rate is 1.3 GHz for the NVIDIA Tesla C1060 [10]. For the Xilinx XC5VLX330T, we used Xilinx Power Estimator (XPE) 12.1 [11]. To consider the power-based performance, we set the process factor as "Maximum" and apply all the related hardware resources in the XPE. The XPE reports 18.8 W for the peak and 5.7 W for the static power consumption. Based on (4), the power-based performances are described in (7).

$$\alpha_{MAX\text{-}GPU} = \{(187.8\text{-}57.7)/1.3\} \times 10^{-9} \approx 10^{-7}$$
$$\alpha_{MAX\text{-}FPGA} = \{(18.8\text{-}5.7)/0.12\} \times 10^{-9} \approx 1.1 \times 10^{-7}$$
$$MAX(F_{WATT\text{-}D\text{-}GPU}) = F_{CLK\text{-}GPU} \times 10^{7}$$
$$MAX(F_{WATT\text{-}D\text{-}FPGA}) = F_{CLK\text{-}FPGA} \times 0.91 \times 10^{7} \quad (7)$$

where, $F_{CLK\text{-}GPU}=\{F_{GPU}/1.3\}\times10^{-9}$, $F_{CLK\text{-}FPGA}=\{F_{FPGA}/1.2\}\times10^{-8}$.

IV. Results

Figures 1 and 2 represent the performance for a mixed precision solver for the GPGPU and the FPGA platform with the notation {a, b, c} for the vertical axis, where *a*, *b*, and *c* are the power, clock, and time-based performance. When condition numbers are small, high performance is obtained since we can apply lower precision.

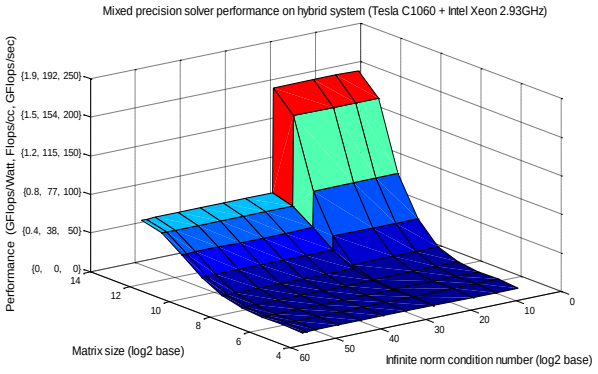

Figure 1. Performance on Intel Xeon 2.93GHz multi-core + Tesla C1060

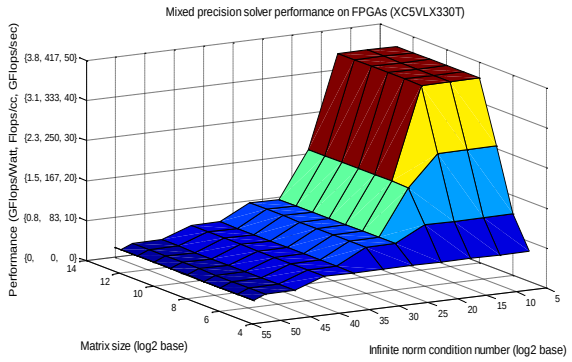

Figure 2. Performance on Xilinx XC4VLX200

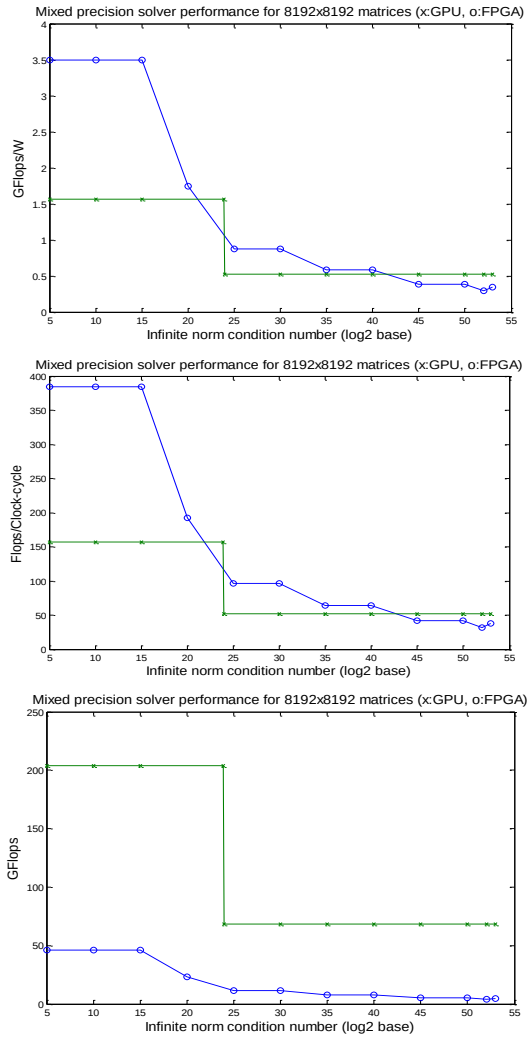

Figure 3. Performance Metric Comparisons

We compare the power-, clock-, and time-based performance according to condition numbers for 8192×8192 matrices in Figure 3. Based on the figure, the FPGA generally shows better performance for the power-based and clock-based performance than the GPGPU while the GPGPU shows better time-based performance.

V. Conclusions

The power-aware performance represents the achievable performance for one additional watt. α_{MAX} values are similar in the case-study between the FPGA and GPGPU. Therefore the clock-based performance almost directly reflects the relative power-based performance. The FPGA shows power-based performance better than the GPGPU in the case-study, since we can obtain higher clock-based performance due to flexibility of the design choices in the FPGA. In addition, a larger FPGA (e.g. Xilinx XC6VSX475T) could improve the performances (power, clock, and time-based) more than 10 times more than the XC5VLX330T since the number of DSP48Es is more than 10 times larger than in the XC5VLX330T.

Acknowledgment

This work was partially supported by the National Science Foundation, grant NSF CHE-0625598.

References

[1] J. L. Gustafson, "Reevaluating Amdahl's law," *C. ACM*, **31(5);** 532-533, 1988.

[2] S. Williams, A. Waterman, and D. Patterson, "Roofline: an insightful visual performance model for multicore architectures," *C. ACM*, **52(4);** 65-76, 2009.

[3] P. Abusaidi, M. Klein, and B. Philofsky, "Virtex-5 FPGA System Power Design Consideration," White Paper 285, Xilinx, 2008.

[4] J. Langou, *et al.*, "Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy (revisiting iterative refinement for linear systems)," Proceedings of the 2006 ACM/IEEE Conference on Supercomputing.

[5] J. Lee, G. D. Peterson, R. J. Hinde, and R. J. Harrison, "Mixed Precision Dense Linear System Solvers High Performance Reconfigurable Computing," *Symposium on Application Accelerators in High Performance Computing 2009.*

[6] J. Sun, G. D. Peterson, and O. O. Storaasli, "High-Performance Mixed-Precision Linear Solver for FPGAs," *IEEE Trans. Comput.*, **57(12)**; 1614-1623, 2008.

[7] J. Lee, *et al.*, "Accelerator performance comparison for mixed precision linear solvers," *IEEE Symposium on Field-Programmable Custom Cumputing Machines* 2010.

[8] J. Demmel, *et al.*, "Error bounds from extra-precise iterative refinement," *ACM Trans. Math. Softw.*, **32(2);** 325-351, 2006.

[9] *MAGMA v0.2 user guide*. Available: http://icl.cs.utk.edu/magma/

[10] *NVIDIA Tesla C1060 Computing Processor*. Available: http://www.nvidia.com/object/product_tesla_c1060_us.html

[11] *Xilinx Power Estimator 12.1*. Available: http://www.xilinx.com/products/design_resources/power_centra

[12] B. Holland, K. Nagarajan, and A. D. George, "RAT: RC Amenability Test for Rapid Performance Prediction," *ACM Trans. Reconfigurable Technol. Syst.*, **1(4);** 1-31, 2009.

[13] M. C. Smith and G. D. Peterson, "Parallel application performance on shared high performance reconfigurable computing resources," *Perform. Eval.*, **60(1-4);** 107-125, 2005.

[14] J. Williams, *et al.*, "Computational Density of Fixed and Reconfigurable Multi-Core Devices for Application Acceleration," in *Reconfigurable Systems Summer Institute (RSSI)*, Urbana, IL, 2008.

[15] J. Williams, *et al.*, "Fixed and Reconfigurable Multi-Core Device Characterization for HPEC," in *High-Performance Embedded Computing Workshop (HPEC)*, Lexington, MA, 2008.