Comparing and Evaluating GPU Platforms with a Single Point

Nitin Satpute*, Roberto Giorgi*,[1],

* *Department of Information Engineering and Mathematics, University of Siena, Italy*

ABSTRACT

We have seen from the literature that it's not immediate to compare performance results from different published research papers. We propose a methodology to summarize typical curves used to indicate the performance of a given computing platform. We present performance of matrix multiplication on a GPU platform using a single point. We form clusters of points with similar performance values. We have found three clusters for the SGEMM, SGEMM & DGEMM, and DGEMM. We compare and evaluate platforms in these clusters based on the underlying hardware platform, precision, algorithm, library, etc.

KEYWORDS: Matrix Multiplication; GPU; Performance Evaluation

1 Introduction

A huge number of papers have been published on comparing different computing platforms. However we realized that is not so immediate to compare results from different research groups or even from same work. Therefore we propose a methodology to summarize typical curves that are used to indicate the performance of a given computing platform. Since nowadays GPUs are one of the most promising high performance computing platforms (also integrated on same die as CPUs in Intel, AMD, NVIDIA chips), we selected them to illustrate our methodology. However, many different applications are used to show the benefits of a computing platfom. We restricted our attention to probably the most widely used benchmark, i.e., Matrix Multiplication (MM) of dense matrices which is a very common kernel in a vast variety of applications [VD08].

Our proposed methodology aims at identifying a single point which could identify a fundamental change in the reference curves we considered. We identify such point for each combination of platform, precision, algorithm, library, etc, and represent those points in a graph in order to compare performance and limitations more easily for the actual benchmark (MM in our case study). The goal is to provide GPU comparison based on MM. The rest of the paper is organized as follows. Section 2 presents performance comparisons of various MM algorithms on different parallel computing platforms in terms of Cluster Analysis. Finally, Section 3 concludes and discusses about future research issues.

[1]E-mail: {satpute,giorgi}@dii.unisi.it

[2]This work is partly funded by the EU projects, HiPEAC (id. 287759) and AXIOM (id. 645496).

2 Cluster Analysis

Cluster analysis is intended to identify group of platforms with comparable performance. We have plotted several points in the graph considering the Last Linear Point. These clusters are made in order to understand the performance variations in different platforms. This permits for example the selection of a platform for expected performance. The goal of this analysis is to

1. Distinguish performance of SGEMM and DGEMM on different platforms.
2. Identify characteristics of MM on different platforms (HW/SW) for clusters of similar performance values.

Let us consider a typical weak scaling curve [VD08]. We define "Last Linear Point" as follows. We identify the linear region (GFlops = K (MS)) where MS is the matrix size (X-axis). For each subsequent point, we measure the vertical distance from the previous linear region. When the deviation of the curve from linear region is larger than given threshold (Th), the point is referred as Last Linear Point (LLP). Figure 1 gives the analysis of LLP. Mathematically,

$$delta = kX - f(X)$$
$$delta > Th$$
$$Th = 5\% * k * X \tag{1}$$

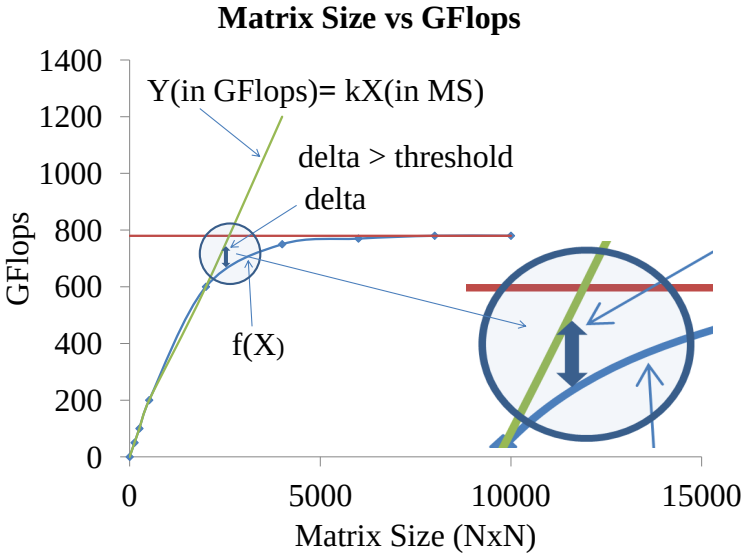
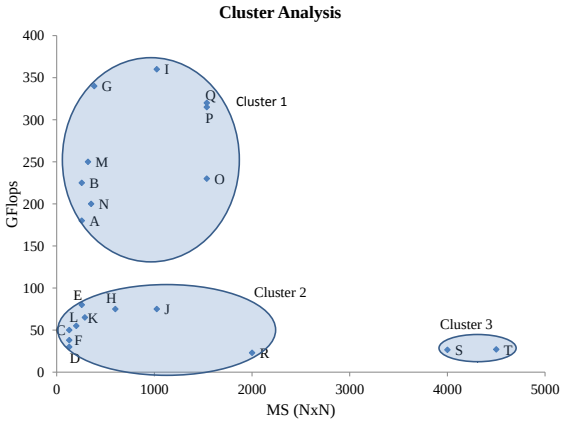


Figure 1: Last Linear Point

One challenging step is the evaluation of the slope k of the linear region. This can be rigourously done by linear regression techiques. (not shown here.) Please note that in this study, we choose 5% in equation 1. Further interpolation of such value is left on future work.

Reasons for calculating the LLP are

1. It can always be calculated.
2. It gives maximum values of MS and GFlops for which performance deviates first from linear region.
3. It can be used to form clusters of similar performance values.



POINTS	MS	GFLOPS	Reference Paper and Figure (f)	GEMM	PLATFORMS
A	256	180	[VD08](f-5)	SGEMM	9800 GTX (16x1.67 GHz)
B	256	225	[VD08](f-5)	SGEMM	8800 GTX (16x1.35 GHz)
C	128	50	[VD08](f-5)	SGEMM	8600 GTS (4x1.45 GHz)
D	128	30	[VD08](f-5)	SGEMM	Core2Duo, 2.67 GHz
E	256	80	[VD08](f-5)	SGEMM	Core2Quad, 3.0 GHz
F	128	38	[VD08](f-6)	DGEMM	Core2Quad, 3.0 GHz
G	384	340	[VD08](f-5)	SGEMM	GTX 280 (30x1.30GHz)
H	600	75	[VD08](f-6)	DGEMM	GTX 280 (30x1.30GHz)
I	1024	360	[NTD11](f-1)	SGEMM	GTX280, CUDA 2.3
J	1024	75	[NTD11](f-1)	DGEMM	GTX280, CUDA 2.3,
K	288	65	[NTD11](f-8)	DGEMM	GTX 280, MAGMA
L	200	55	[NTD11](f-8)	DGEMM	GTX 280, Cublas 2.3
M	320	250	[NTD11](f-9)	SGEMM	GTX 280, MAGMA
N	352	200	[NTD11](f-9)	SGEMM	GTX 280, Cublas 2.3
O	1536	230	[CCM09](f-5)	SGEMM	GTX 280, Cublas 1.0
P	1536	315	[CCM09](f-5)	SGEMM	GTX 280, Cublas 2.0
Q	1536	320	[CCM09](f-5)	SGEMM	GTX 280, Algo_Paper
R	2000	23	[BB15](f-1)	DGEMM	NE, MKL
S	4000	26.5	[BB15](f-1)	DGEMM	NE, Strassen
T	4500	27	[BB15](f-1)	DGEMM	NE, D'Alberto Strassen-Winograd

(** NE: 2x12 Core Intel 2.4 GHz, Ivy Bridge, 64 GB = Single Compute node on NERSC Edison)

Figure 2: ClusterAnalysis

In figure 2, we compare LLP of several platforms. We found following clusters from the graph.

1. Cluster 1: It is the biggest cluster. It contains points O [CCM09], P [CCM09], Q [CCM09], I [NTD11], G [VD08], M [NTD11], B [VD08], N [NTD11] and A [VD08] displaying the performance results of SGEMM.
2. Cluster 2: It is the second biggest cluster. It contains points showing performance results of SGEMM and DGEMM. Points C [VD08], D [VD08], E [VD08] show the performance results for SGEMM while remaining points F [VD08], L [NTD11], K [NTD11], H [VD08], J [NTD11], and R [BB15] show the performance of DGEMM. The reason for degradation of perfomance of SGEMM (Points C, D, E) is due to the selection of platform parameters and algorithm.
3. Cluster 3: It is the smallest cluster. It contains points S [BB15] and T [BB15] related to the performance results of DGEMM.

From above graphs we can conclude that

1. Double precision MM typically achieves lower performance in the range of 0-100 GFlops.
2. Single precision MM typically achieves higher performance in the range of 200-400 GFlops.
3. The performance of SGEMM or DGEMM increases with increase in the core utilization.

However some exceptions exist that can be seen from fig. 2.

3 Conclusion and Future Work

In this work, we present a relativey simple methodology to clarify and compare computing platforms. We illustrate this technique in the case of GPU(s) and MM. The work will be extended to for the platforms and benchmarks in order to show a systematic approach for performance evaluation. Future research work is required in the following problem areas. We must check the parameters effecting the performance in GPU environments. Therefore, it is necessary to evaluate performance in various GPU environments. We have to write CPU and GPU programs independently when we want to execute parallel programs using CPUs and GPUs. However, it is desirable that users are not concerned about whether they use CPUs or GPUs. Various applications are speeded up when computations to CPUs and GPUs are directly assigned in parallel processing. Utilizing multiple processor environment will become a new trend in GPU technology for the benefit of many CPUs and GPUs. In such environments, new approaches for realizing optimal load balancing are required to achieve the maximal speed up in the high-performance computing field.

References

[BB15] Austin R. Benson and Grey Ballard. A framework for practical parallel fast matrix multiplication. In *Proceedings of the 20th ACM SIGPLAN Symposium on Principles and Practice of Parallel Programming*, PPoPP 2015, pages 42–53, New York, NY, USA, 2015. ACM.

[CCM09] Xiang Cui, Yifeng Chen, and Hong Mei. Improving performance of matrix multiplication and fft on gpu. In *ICPADS*, pages 42–48. IEEE, 2009.

[NTD11] Rajib Nath, Stanimire Tomov, and Jack Dongarra. Accelerating gpu kernels for dense linear algebra. In *Proceedings of the 9th International Conference on High Performance Computing for Computational Science*, VECPAR'10, pages 83–92, Berlin, Heidelberg, 2011. Springer-Verlag.

[VD08] Vasily Volkov and James W. Demmel. Benchmarking gpus to tune dense linear algebra. In *Proceedings of the 2008 ACM/IEEE Conference on Supercomputing*, SC '08, pages 31:1–31:11, Piscataway, NJ, USA, 2008. IEEE Press.