Do theoretical FLOPs matter for real application's performance ?

Joshua.Mora@amd.com

Abstract: The most intelligent answer to this question is "it depends on the application". To proof that, we will show a few examples from both theoretical and practical point of view. In order to validate experimentally it, a modified AMD processor named "Fangio" (AMD Opteron 6275 Processor) will be used which has limited floating point capability to 2 FLOPs/clk/BD unit, delivering less (-8% in avergage) but close to the performance of AMD Opteron 6276 Processor with 4 times more floating point capability , ie. 8 FLOPs/clk/BD unit.

The intention of this work is threefold: i) to demonstrate that the FLOPs/clk/core of microprocessor architectures isn't necessarily a good performance metric indicator despite it is heavily used by the industry (eg. HPL). ii) to expose that code vectorization technology of compilers is fundamental in order to extract as much as possible real application performance but it has a long way to go in extracting it. iii) It would not be fair to blame exclusively on compiler technology: algorithms are not well designed and written for the compilers to exploit vector instructions (ie. SSE, AVX and FMA).

HPC Advisory Council , Malaga, Spain, 2012

Agenda

- Concepts
  - Kinds of FLOPs/clk
- AMD Interlagos processor FPU
  - FPU, see *Understanding Interlagos arch through HPL* (HPC Advisory Council workshop, ISC 2012)
  - Roofline model
- AMD Fangio processor
  - FPU capping, roofline model
- Results/Conclusions within roofline model for Interlagos and Fangio.
  - Benchmarks: HPL, stream, CFD apps.

Concepts: kinds of FLOPs/clk

- A brief list of floating point instructions and examples supported by AMD Interlagos
- Scalar, Packed
- SP: Single precision, DP: Double precision

Ins. Type	Examples	FLOPs/clk	Reg. size
X87	FADD , FMUL	1	32,64bits
SSE (SP)	Scalar: ADDSS , MULSS Packed: ADDPS,MULPS	8	128bits
SSE2 (DP)	Scalar: ADDSD, MULSD Packed: ADDPD,MULPD	4	128bits
AVX (SP)	Scalar: VADDSS, MULSS Packed: VADDPS, VMULPS	4, 8	128,256b
AVX (DP)	Scalar: ADDSD, MULSD Packed: VADDPD, VMULPD	2, 4	128,256b
FMA4 (SP)	Scalar: VFMADDSS Packed: VFMADDPS	8, 16	128,256b
FMA4 (DP)	Scalar: VFMADDSD Packed: VFMADDPD	4, 8	128,256b

Roof line for AMD Interlagos

System: 2P, 2.3GHz, 16 cores, 1600MHz DDR3.

Real GFLOP/s in double precision (Linpack benchmark)

2procs x 8core-pairs x 2.3GHz x 8 DP FLOP/clk/core-pair x 0.85eff =

250 DP GF/s

Very high numerical intensity ie. (FLOP/s) / (Byte/s)

- **use of AMD Core Math Library (FMA4 instructions)**
- **cache friendly**
- **reusability of data**
- **DGEMM is L3 BLAS has Arithmetic Intensity order N (problem size)**

Real GB/s: 72 GB/s (stream benchmark)

Low numerical intensity

- **use of non temporal stores (use of write combined buffer instead of evicting data into L2 -> L3 -> RAM to speed up the write into RAM.)**
- **not cache friendly**
- **no reusage of data**
- **most of the time cores waiting for data (low FLOP/clk despite using SSE2)**
- **stream is L1 BLAS has Arithmetic intensity order 1 (size independent)**

AMD Fangio, FPU capping

- Fangio is Interlagos processor model 6276 but has capped FPU from 8 DP FLOP/clk to 2 DP FLOP/clk by slowing down FPU retirement of instructions.
- Allows same instruction architecture set as Interlagos.
- System: 2P, 2.3GHz, 16 cores, 1600MHz DDR3.

Performance impact depends on workload:

Real GFLOP/s in double precision (Linpack benchmark)

2procs x 8core-pairs x 2.3GHz x 2 DP FLOP/clk/core-pair x 0.85eff = 75 DP GF/s

Real GB/s: 72 GB/s (stream benchmark)

unmodified memory throughput performance !

HPL runs to confirm FPU capping

2P 16cores @ 2.3GHz (6276 Interlagos)

```
================================================================================
T/V                N      NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01R2L4       86400   100     4     8            1774.55              2.423e+02
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0022068 ..... PASSED
================================================================================
```

2P 16cores @ 2.3GHz (6275 Interlagos) Fangio

3x longer time

```
================================================================================
T/V                N      NB     P     Q               Time                 Gflops
--------------------------------------------------------------------------------
WR01R2L4       86400   100     4     8            5494.39              7.826e+01
--------------------------------------------------------------------------------
||Ax-b||_oo/(eps*(||A||_oo*||x||_oo+||b||_oo)*N)=        0.0022316 ..... PASSED
================================================================================
```

Stream runs on 6275 to confirm no drop in memory throughput

Function	Rate (MB/s)	Avg time	Min time	Max time
Copy:	73089.4045	0.0443	0.0438	0.0449
Scale:	68952.3038	0.0469	0.0464	0.0472
Add:	66289.3072	0.0729	0.0724	0.0734
Triad:	**66301.0957**	0.0730	0.0724	0.0734

Scale, Add and Triad do FLOPS in double precision.

Triad is plotted in roofline model since it is the one with highest FLOPs associated with operations: add and multiply using FMA4.

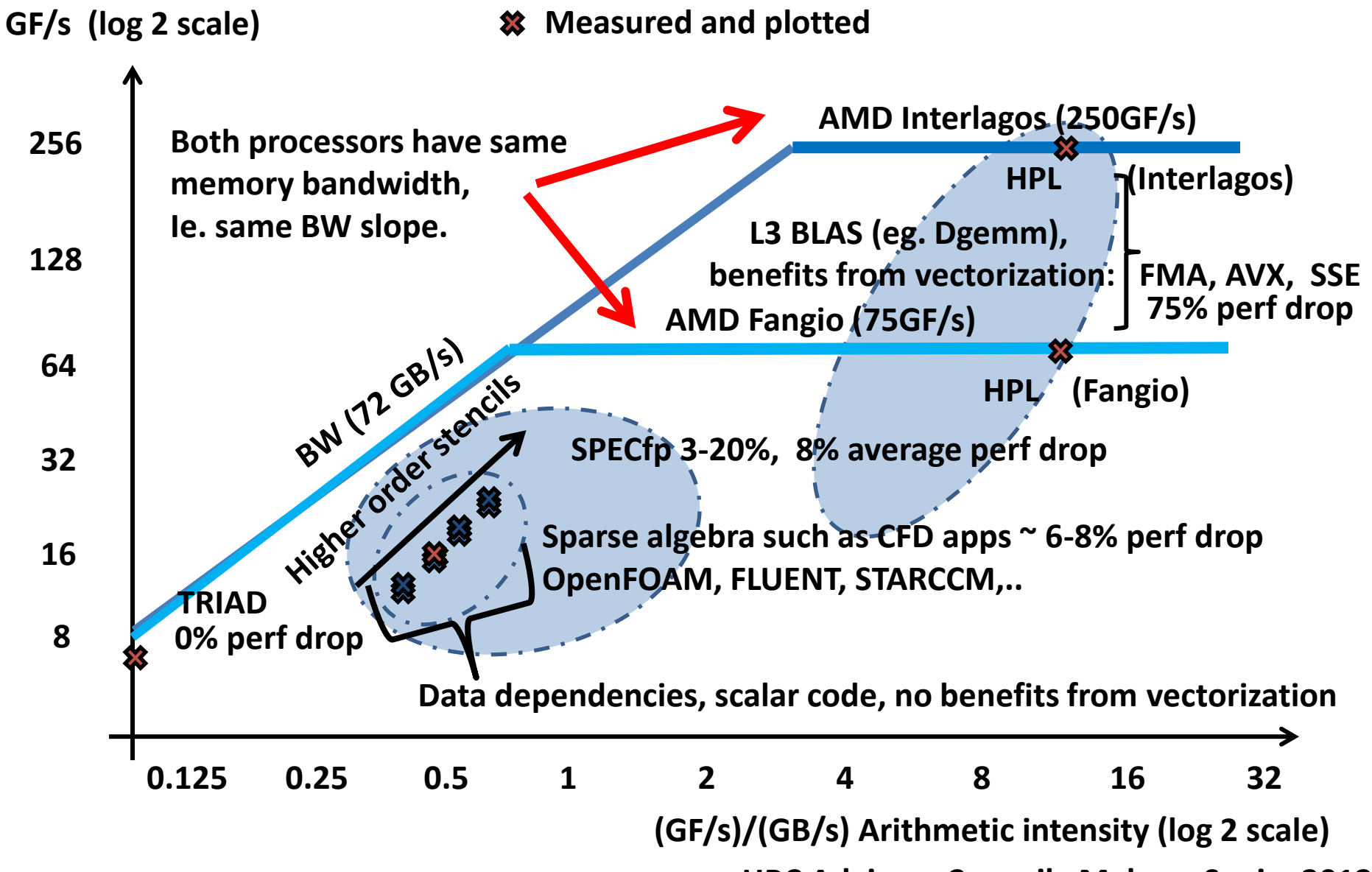
```
#pragma omp parallel for
        for (j=0; j<N; j++)  a[j] = b[j]+scalar*c[j];
```

Summary of measurements per node plotted in roofline model

Workload	Interlagos 6276			Interlagos 6275 "Fangio"		
	GB/s	DP GF/s	AI= F/B	GB/s	DP GF/s	AI= F/B
HPL	=**6***4=24	=**7.8***32=250	10.6	=**1.8***4=6.8	=**2.3***32=75	11.7
STR. TRIAD	=**17***4=68	=**0.5***32=16	0.08	=**17***4=68	=**0.5***32=16	0.08
OPENFOAM	=**15***4=60	=**0.8***32=25	0.41	=**14***4=56	=**0.7***32=22	0.39

- 1 computer has 2 processors with a total of 4 numanodes and 32 cores in 16 compute units
- 1 numanode has a total of 4 compute units.
- Memory bandwidth in GB/s is measured per numanode. **(in red)**
- Double precision floating point is measured per core.**(in red)**

Roofline for Interlagos and Fangio



Performance impact on CFD apps

- **Most of CFD apps with Eulerian formulation use sparse linear algebra to represent the linearized Navier-Stokes equations on non structured grids.**
- **The higher the discretization schemes, the higher the arithmetic intensity**
- **Data dependencies in both spatial and time prevent vectorization**
- **Large datasets have low cache reutilization.**
- **Cores are waiting most of the time to get new data into caches.**
- **Once data is on the caches, the floating point instructions are mostly scalar instead of packed.**
- **Compilers have hard time in finding opportunities to vectorize loops.**
- **Loop unrolling and partial vectorization of independent data help very little due to cores waiting to get that data.**
- **Overall, low performance from FLOPs/s point of view.**
- **Therefore, capping FPU in terms of FLOPs/clk does not impact on application's performance.**
- **Theoretical FLOP/s isn't therefore a good indicator of how applications such as CFD ones (and many more) will perform.**