Usability of the Cache Aware Roofline Model on Knight Landings

Nicolas Denoyelle
nicolas.denoyelle@inria.fr

28 mars 2017

Machine Model

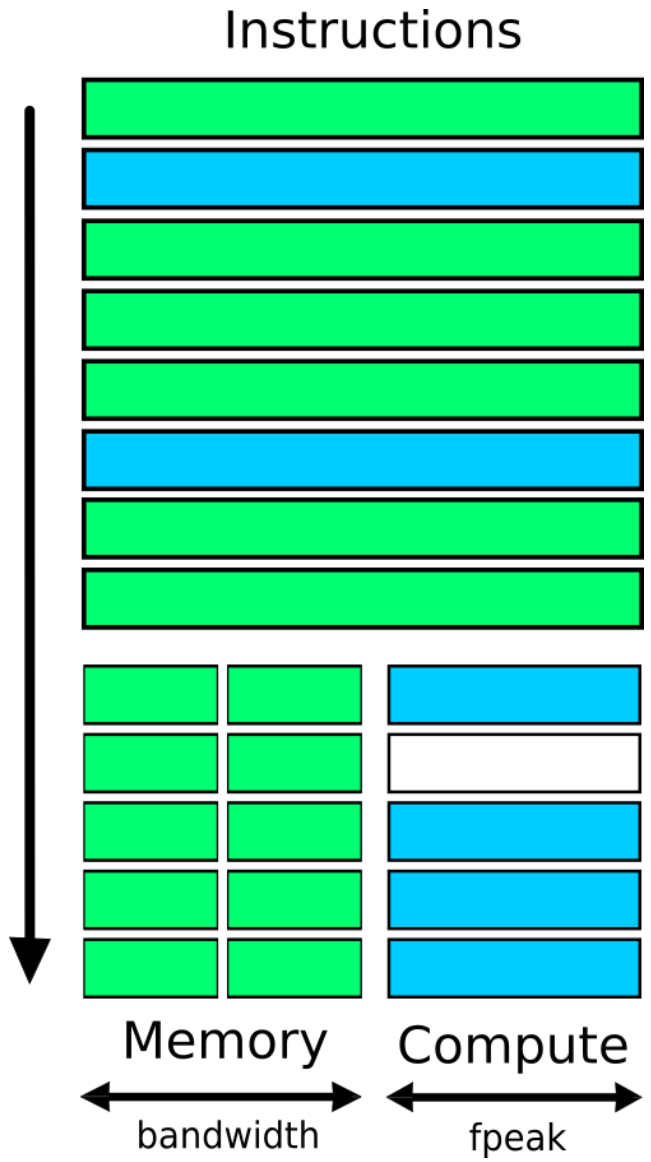


FIGURE – Machine with 2 components

Roofline Model

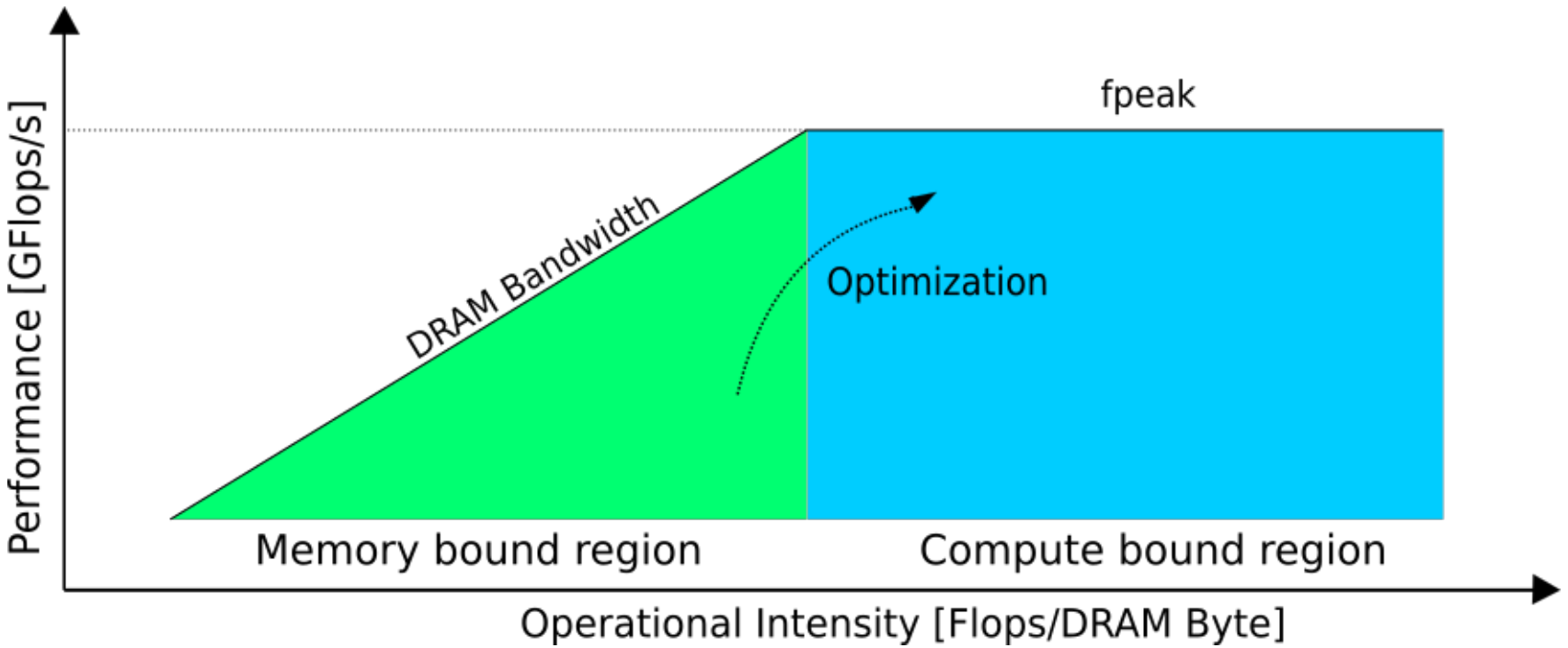


FIGURE – Roofline Model of an hypothetical system with one memory and one compute unit

Enhanced Machine Model

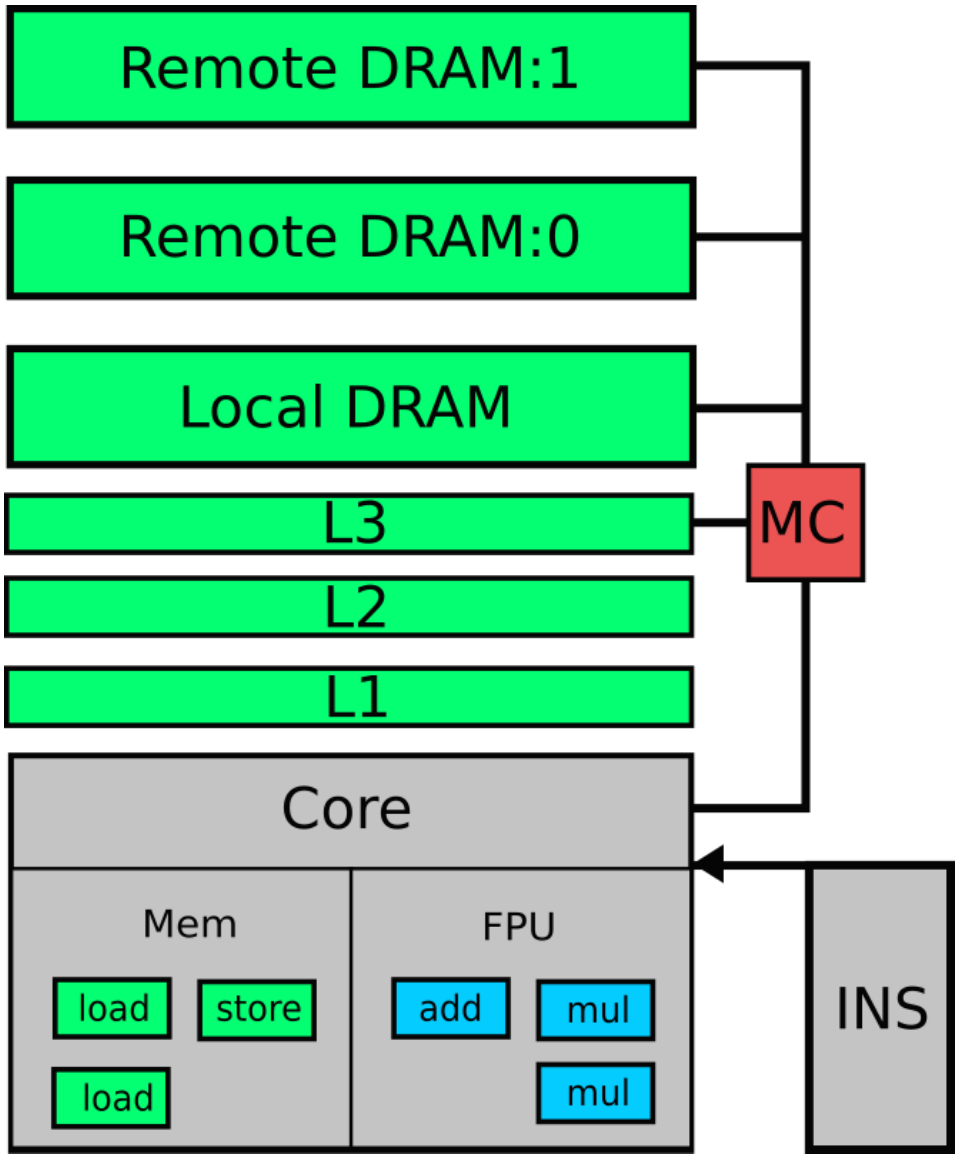


FIGURE – Hypothetical NUMA system with a memory hierarchy and one Core

Cache Aware Roofline Model

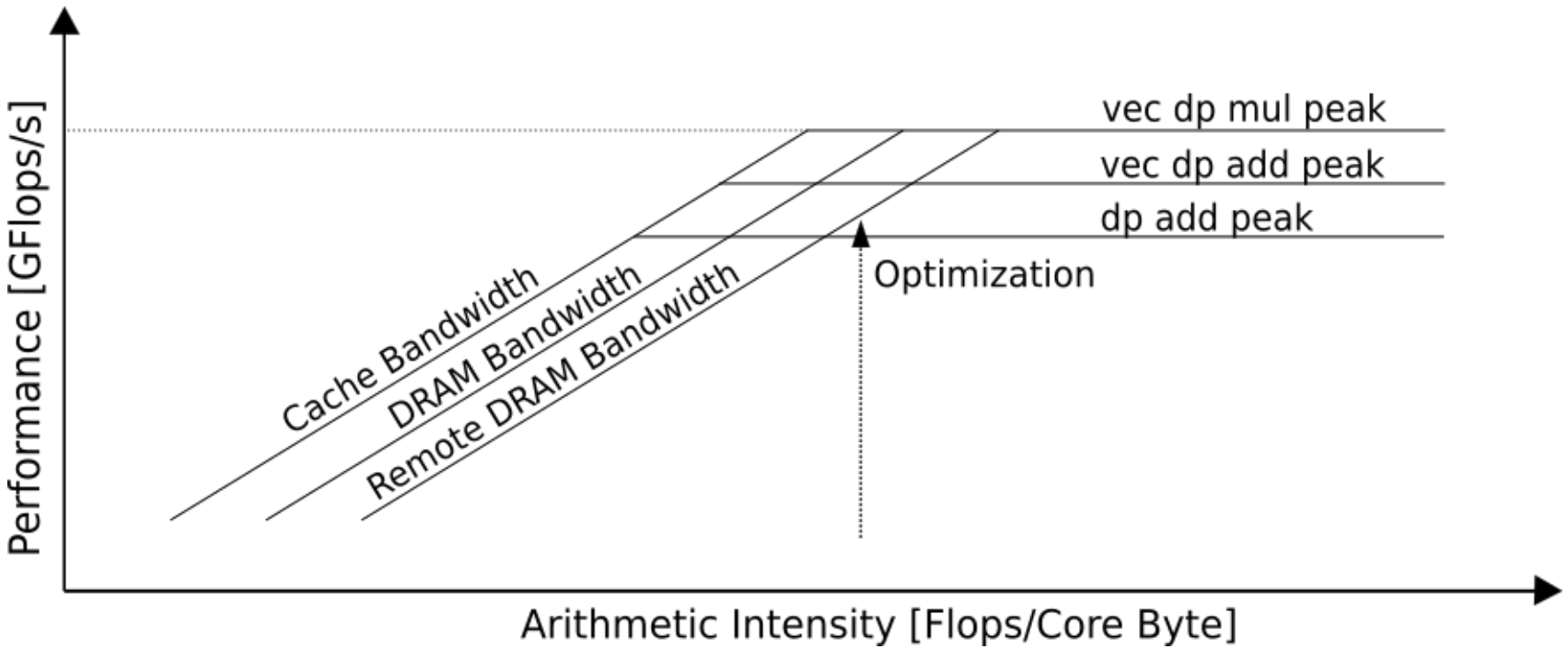
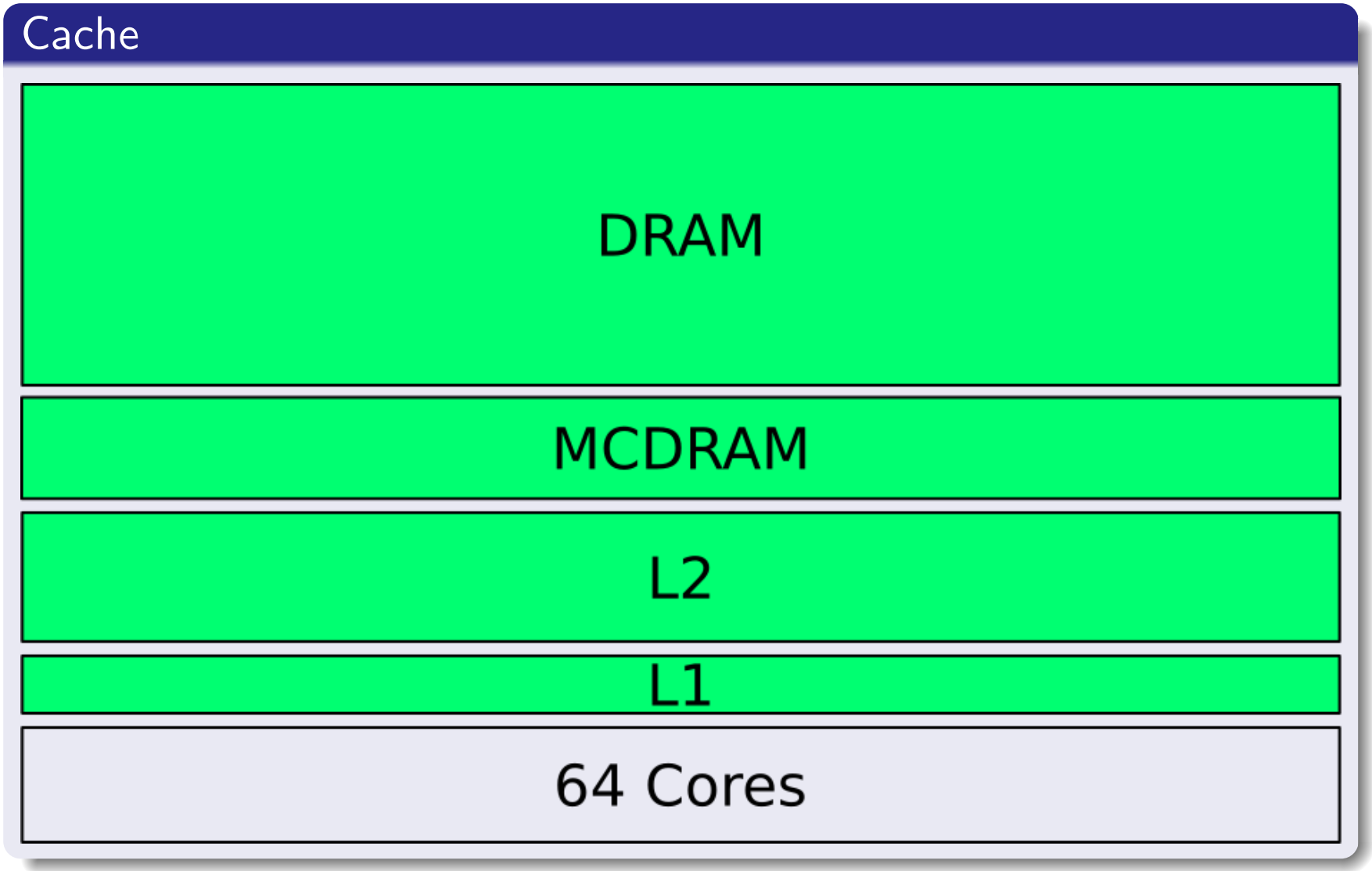


FIGURE – Cache Aware Roofline Model of hypothetical NUMA system with a memory hierarchy and one Core

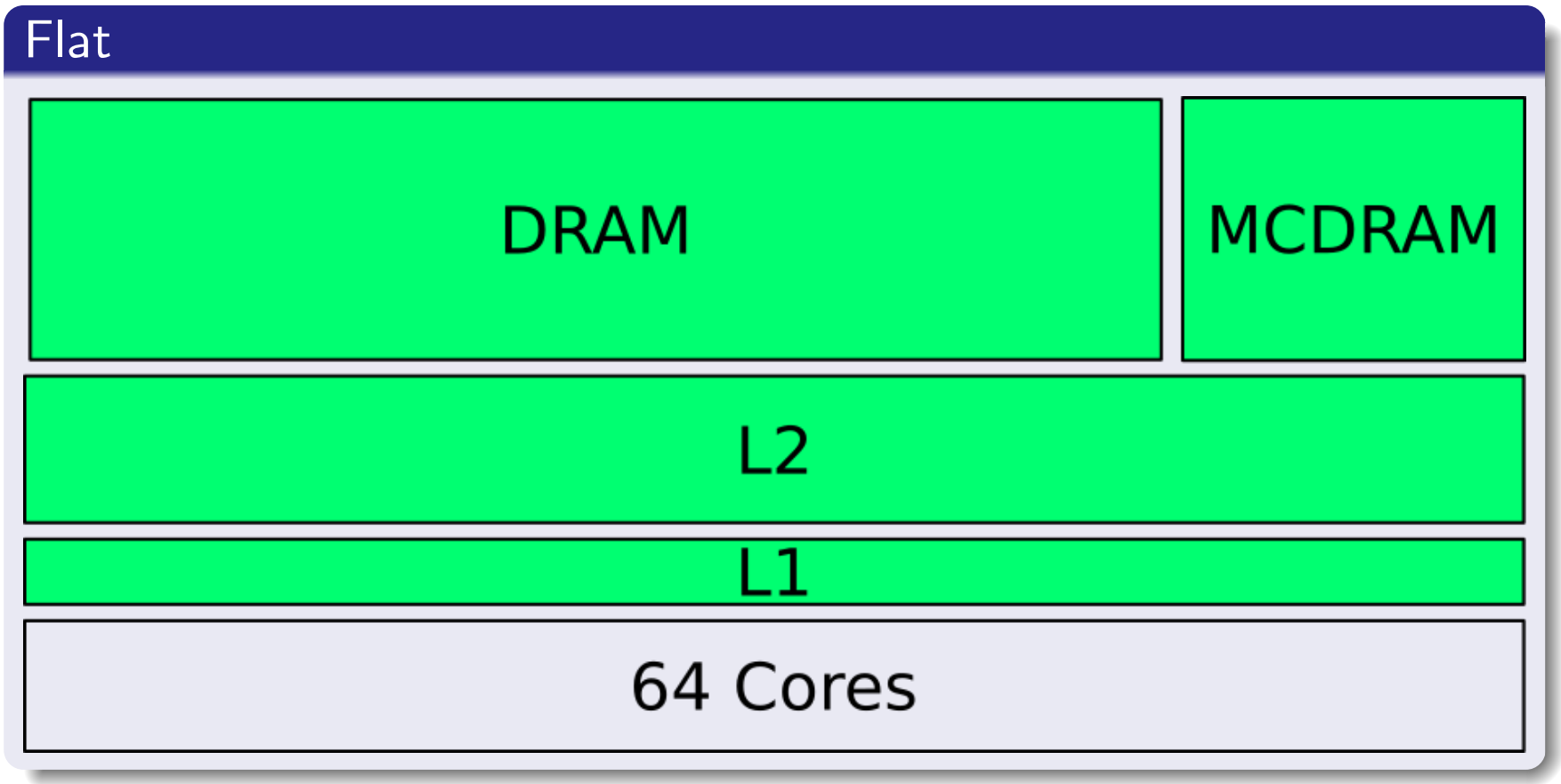
KNL Model

Several modes, several performances (Cf Ian Masliah Talk), one model.



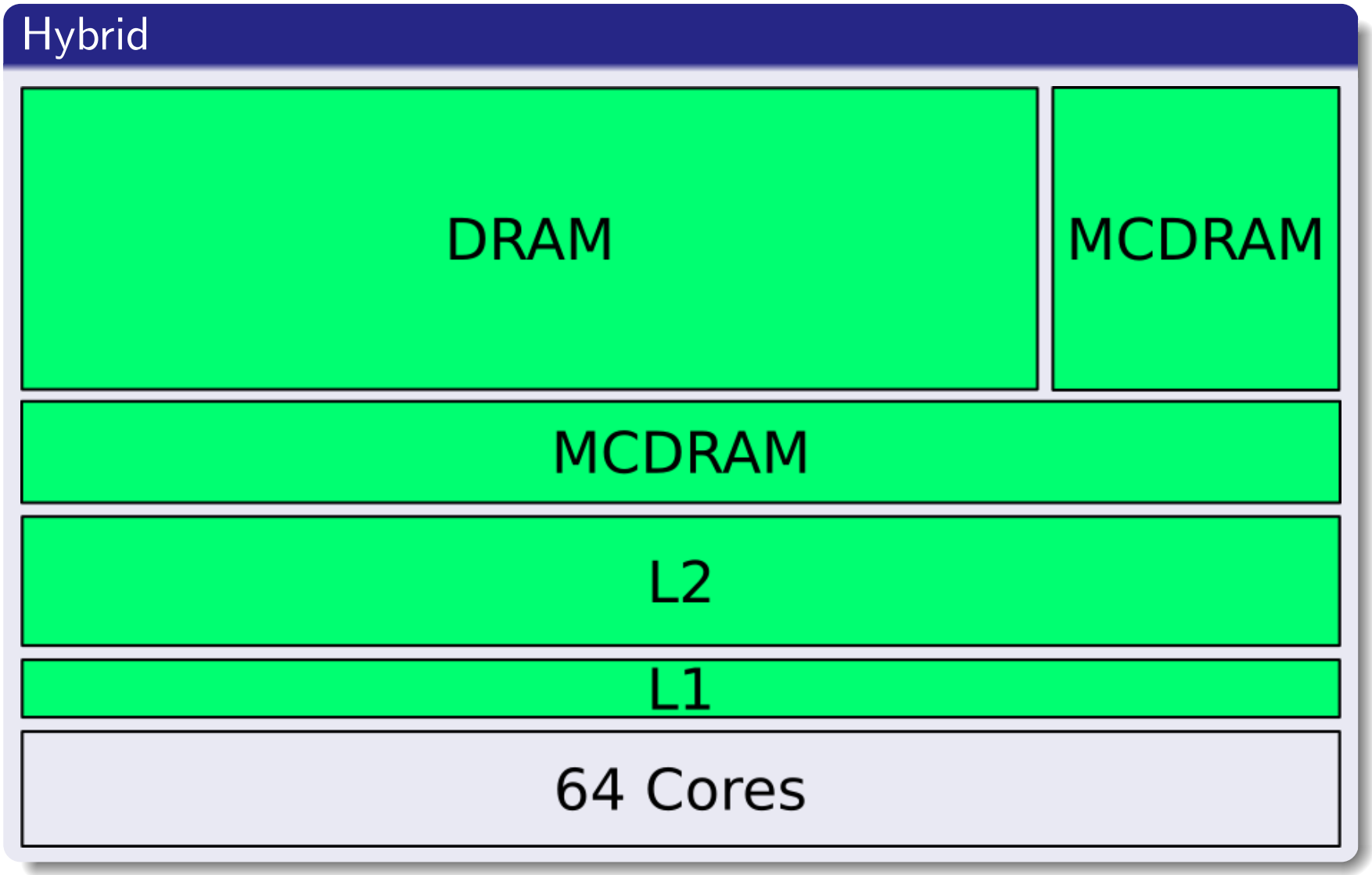
KNL Model

Several modes, several performances (Cf Ian Masliah Talk), one model.



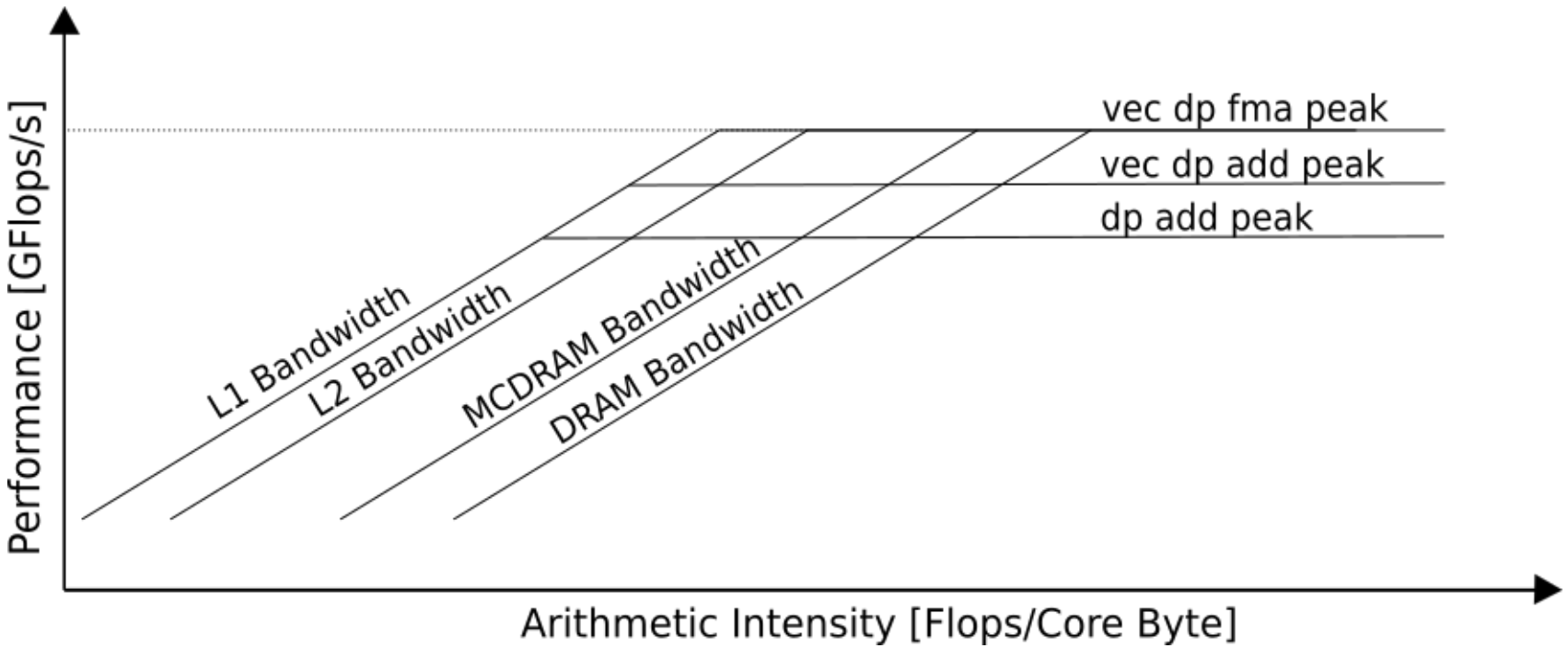
KNL Model

Several modes, several performances (Cf Ian Masliah Talk), one model.



KNL Model

Several modes, several performances (Cf Ian Masliah Talk), one model.



KNL benchmarks (64 threads)

Bandwidths

obj	type	GByte.s	sd
L1	Load	8364	785
L1	Intel	4030	NA
L2	Load	2293	370
L2	Intel	984	NA
MCDRAM	Load	356	7.61
MCDRAM	Store(nt)	267	0.40
MCDRAM	Intel	463.7	NA
DRAM	Load	86.7	0.29
DRAM	Store(nt)	50.9	0.04
DRAM	Intel	83.1	NA

Throughput max $\simeq 1.56$
Instructions/cycle (per core)

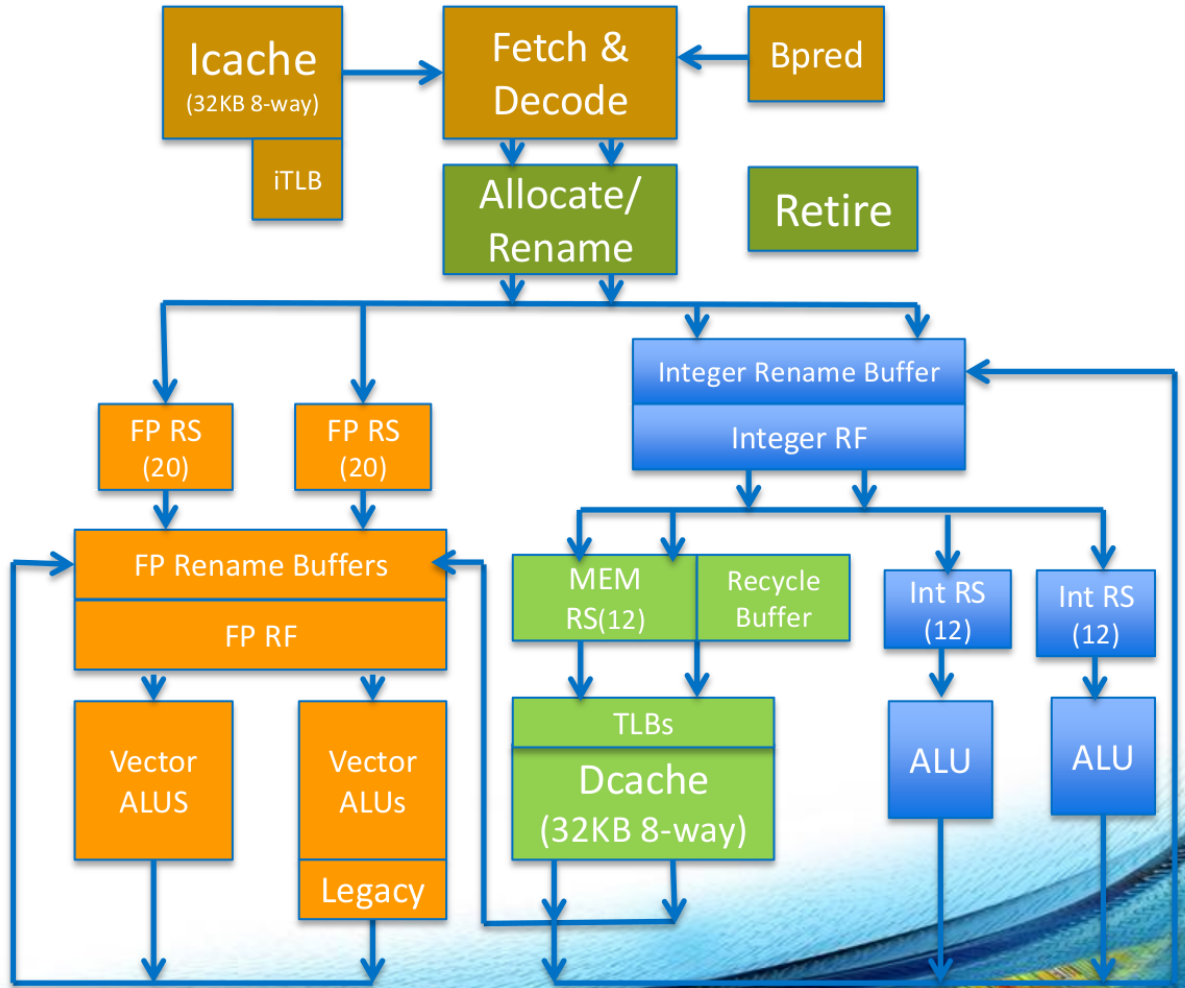
Performance peaks

type	GFlop.s	sd
ADD	1299	8.75
Intel ADD	528	NA
MUL	1299	5.82
FMA	2597	12.6
Intel FMA	1055	NA

Throughput max $\simeq 1.70$
Instructions/cycle (per core)

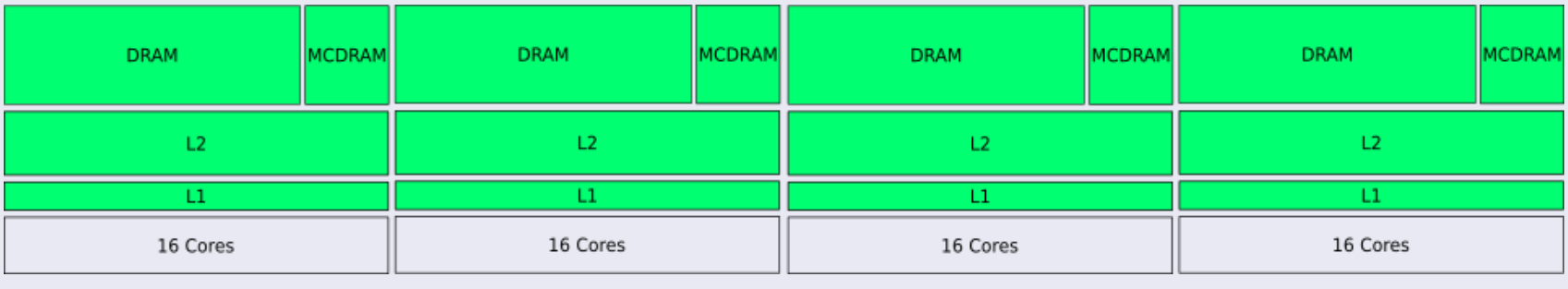
Core & VPU

- 2x 64B Load & 1 64B Store ports in Dcache.



KNL NUMA Bandwidths

SNC4



		from							
		Cluster :0		Cluster :1		Cluster :2		Cluster :3	
		NUMA :0	MCDRAM :1	NUMA :2	MCDRAM :3	NUMA :4	MCDRAM :5	NUMA :6	MCDRAM :7
to	Cluster :0	35.3	65.8	20.8	85.7	34.6	64.5	20.6	70.5
	Cluster :1	35.2	62.6	20.8	86.8	35.6	78.0	20.7	85.8
	Cluster :2	34.9	59.2	20.8	77.8	35.8	86.8	20.8	79.2
	Cluster :3	35.0	60.5	20.7	84.4	35.6	79.0	20.7	86.6

TABLE – Per-cluster load bandwidth (GByte/s) matrix of the KNL.

CARM with Intel Advisor on PlaFRIM

Compile test code with -g

```
module load compiler/intel/64/2017_update2-knl
```

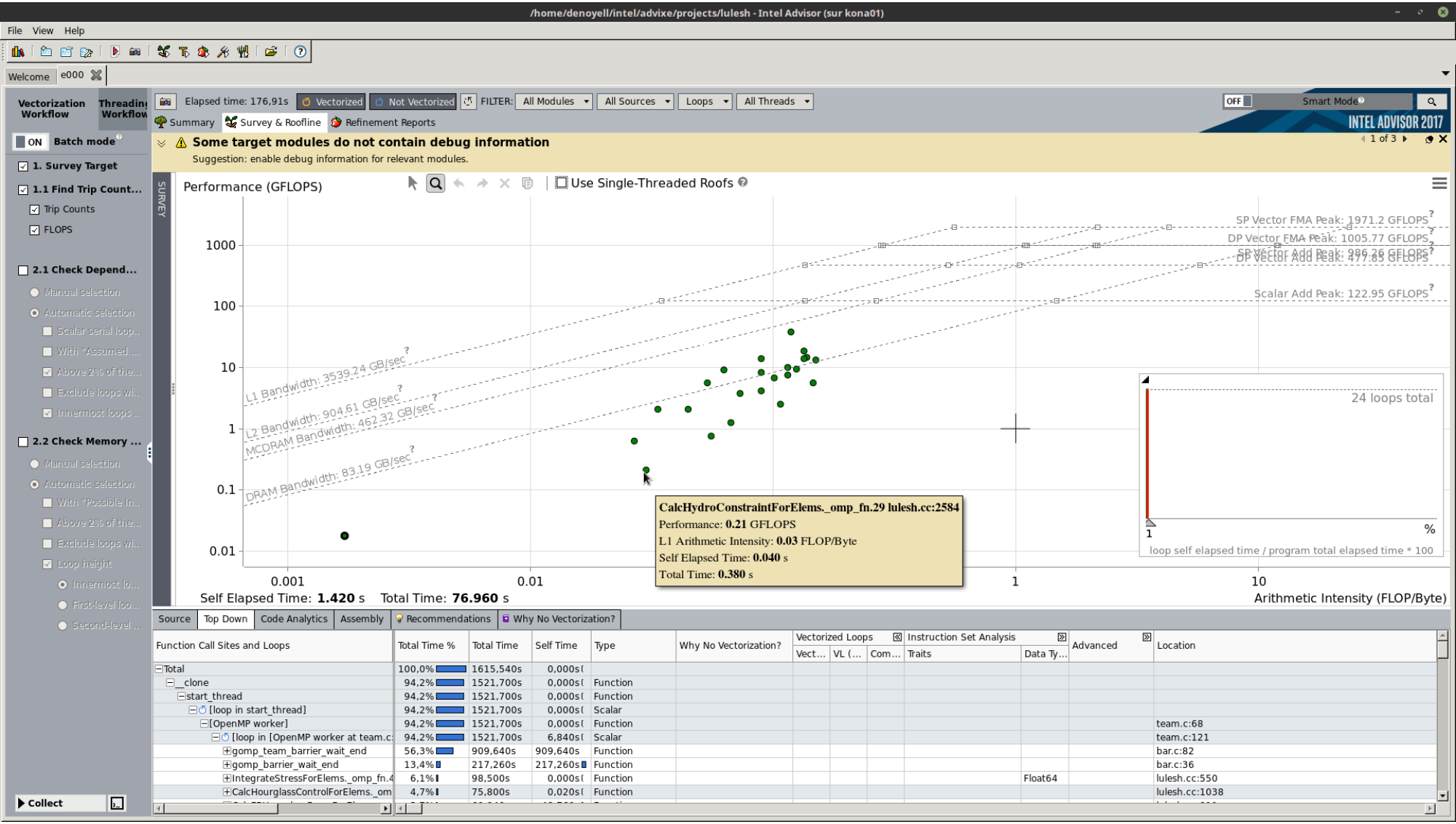
```
source /cm/shared/apps/intel/composer_xe/2017_update2-knl/
advisor_2017.1.2.501009/advixe-vars.sh
```

```
advixe-gui
```

2 runs :

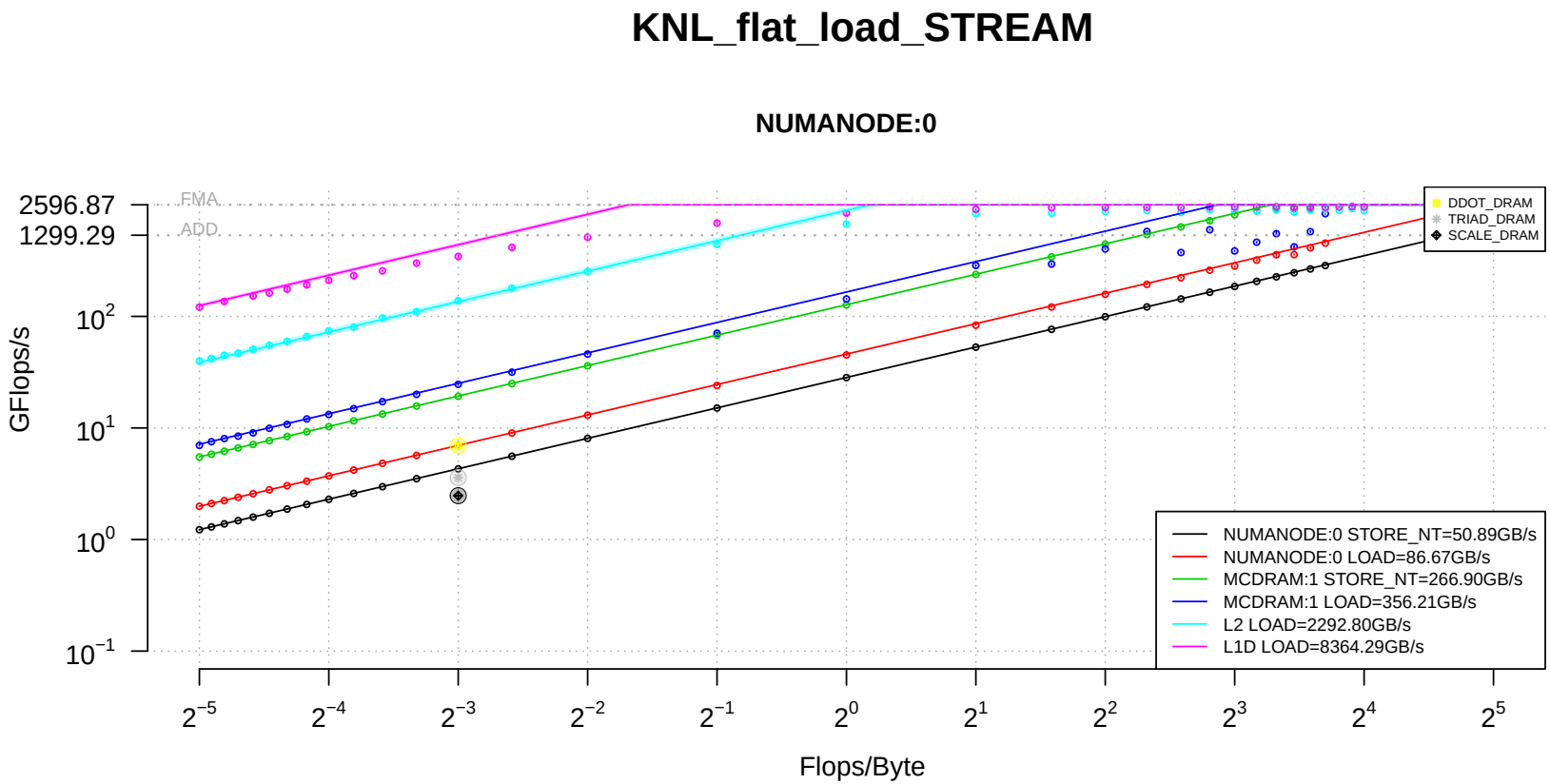
- Instrument code and collect functions, loops : Flops, Bytes.
- Run normal code and collect functions, loops : runtime.

CARM with Intel Advisor on PlaFRIM



Streaming Benchmarks

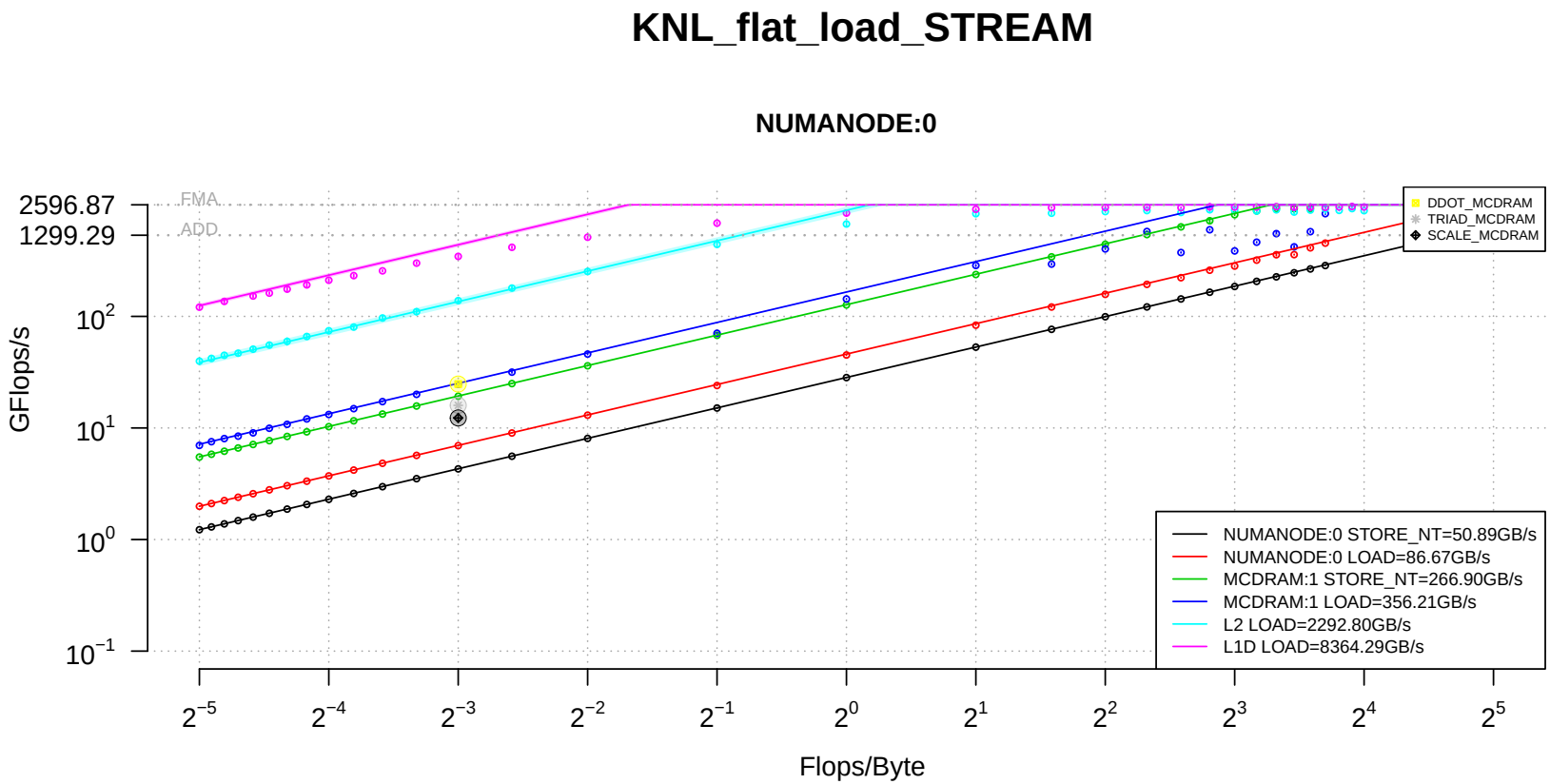
- ddot dot += a[i] * b[i] (2LD + 2 FLOPS)
- scale a[i] = scalar*b[i] (1LD + 1ST + 1 FLOPS)
- triad c[i] = a[i]+scalar*b[i] (2LD + 1ST + 2 FLOPS)



Bandwidth-bound benchmarks below the roof ? !

Streaming Benchmarks

- ddot dot += a[i] * b[i] (2LD + 2 FLOPS)
- scale a[i] = scalar*b[i] (1LD + 1ST + 1 FLOPS)
- triad c[i] = a[i]+scalar*b[i] (2LD + 1ST + 2 FLOPS)



Bandwidth-bound benchmarks below the roof ? !

Analytical model

- s the size of b array and a array, respectively stored and loaded in the process,
- t the wall time,
- B_s the memory store bandwidth,
- B_l the memory load bandwidth,

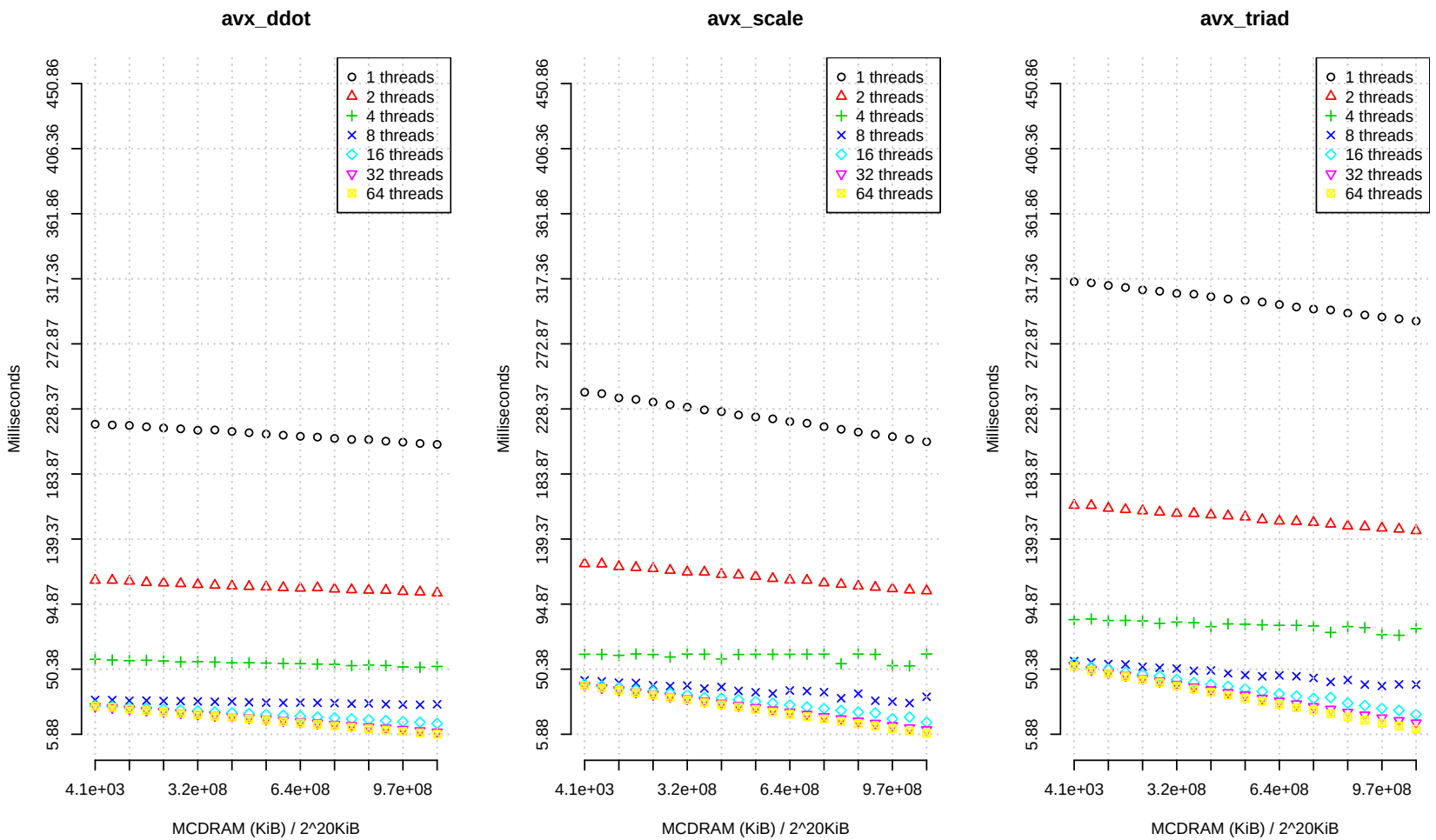
$$t = \frac{s}{B_s} + \frac{s}{B_l} \tag{1}$$

Actually more bandwidths : load, store, store(nt) for each Cluster, for each DRAM and MCDRAM.

$$t = \sum_{\{(s_k, B_l)\}} \frac{s_i}{B_j} \tag{2}$$

Analytical model

Check with DRAM, and MCDRAM.
We allocate 1GB buffer with a slice in DRAM and another in MCDRAM.



Future Works

- CARM for locality, with automatic building and validation.
- Energy CARM.
- Extended analytical model for memory partitionning.
- What about latency ? Only 10% improvement on lulesh proxy-application between DRAM and MCDRAM.

Merci