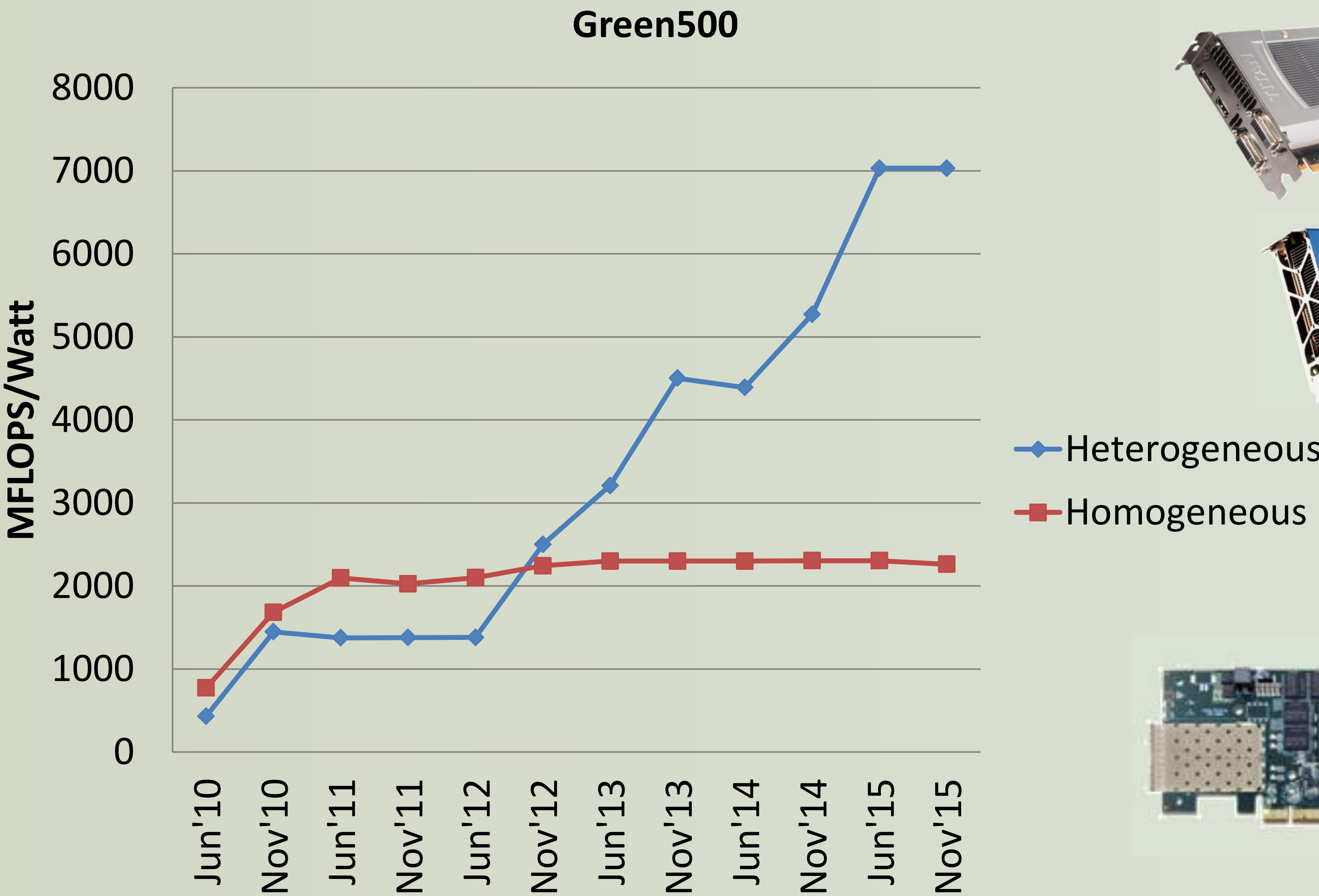
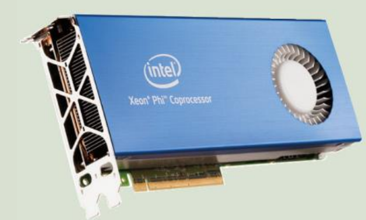
PeaPaw: Performance and Energy Aware Workload Partitioning on Heterogeneous Platforms

Li Tang
Advisor: X. Sharon Hu
University of Notre Dame

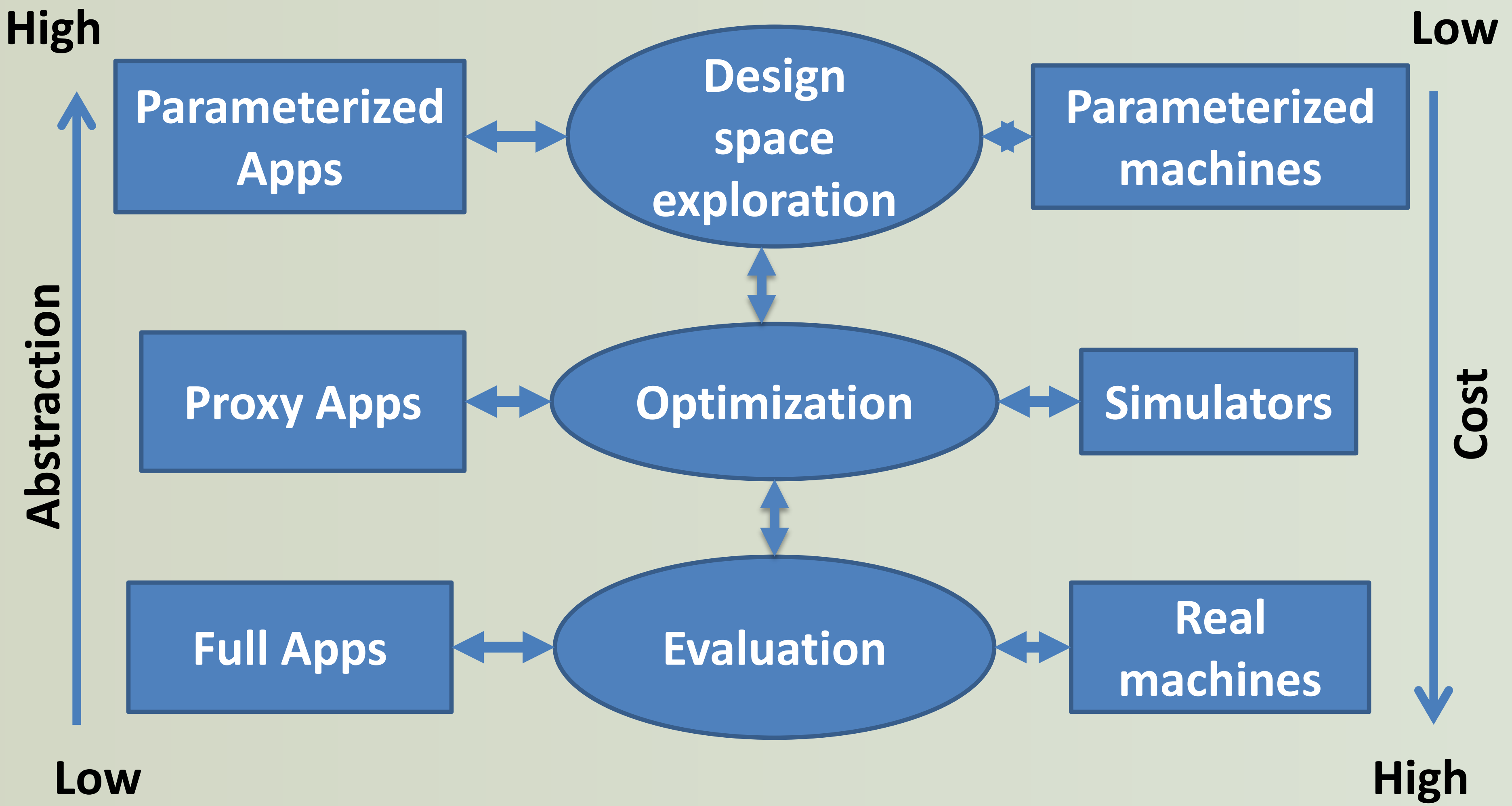
UNIVERSITY OF NOTRE DAME

Energy efficiency





Challenges of heterogeneous computing



PeaPaw Framework

What we want to achieve

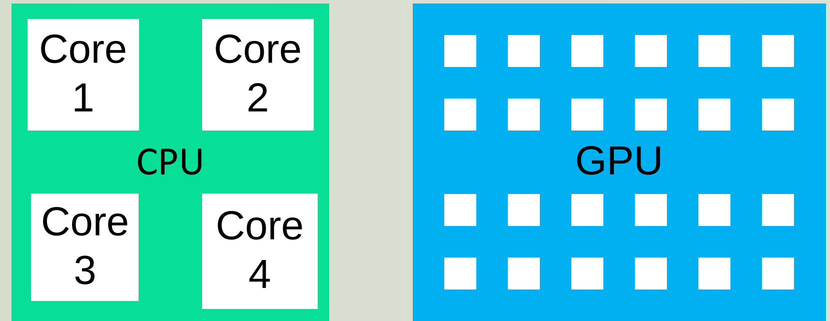
- Given a CPU+GPU platform
  - Find the best workload partition between CPU and GPU
  - Achieve high performance or energy efficiency

	Design				
	Algorithm			Dataset	
Uniprocessor	Algorithm			Data Structure	
Homogeneous	Algorithm	Communication & Scheduling		Data Structure	Dataset Partitioning
Heterogeneous	Algorithm	Communication & Scheduling	Algorithm Partitioning	Data Structure	Dataset Partitioning

Workload partitioning

- Workload
  - Amount of computation (# of flops)
  - Amount of memory traffic (# of bytes)
- Workload partition (WP)
  - Dataset partitioning (DP)
    - Pros: balanced CPU/GPU workload distribution
    - Cons: low execution efficiency
  - Code partitioning (CP)
    - Pros: high execution efficiency
    - Cons: high development cost

Examples of CP and DP



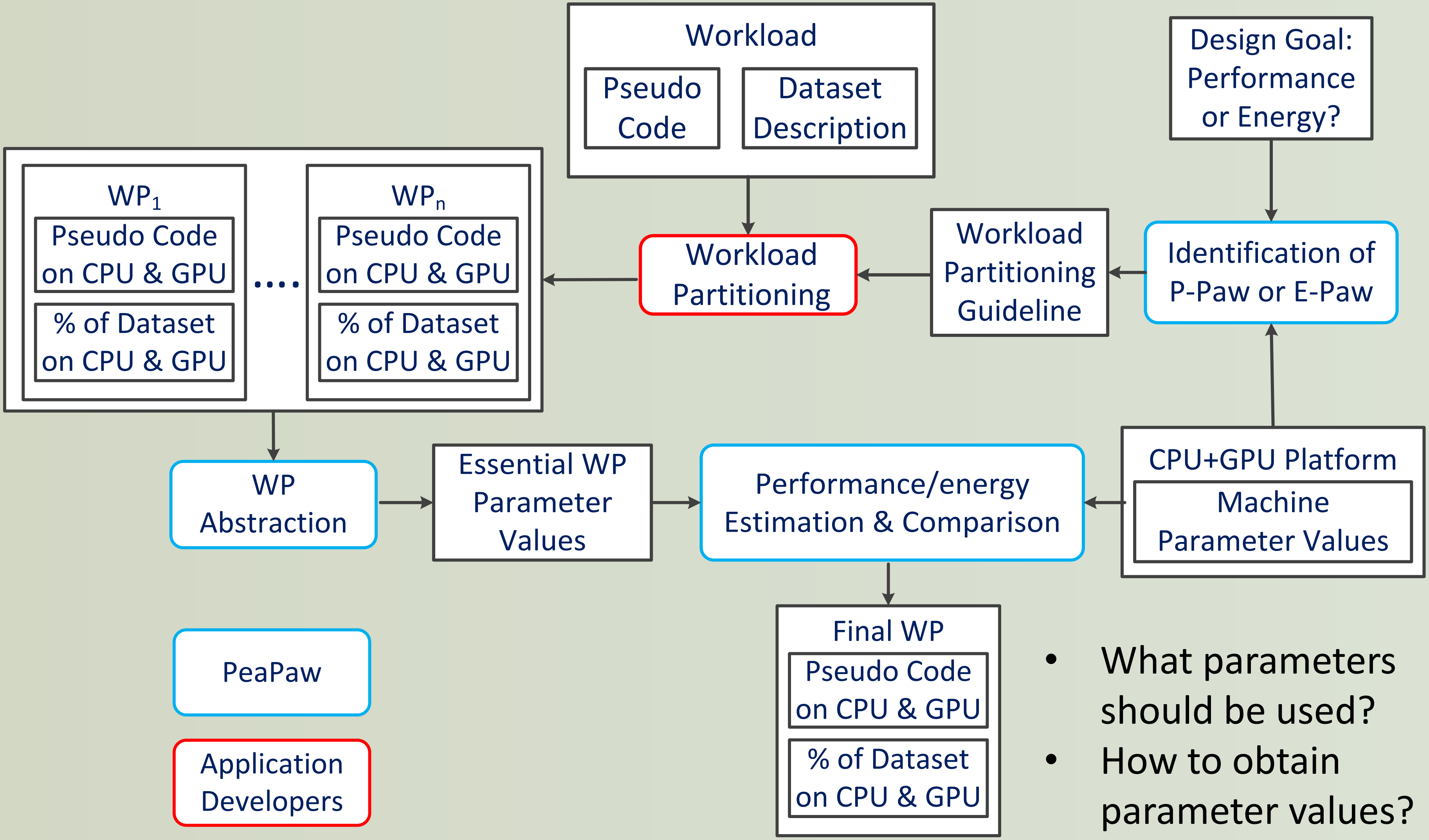

```
For (i=0;i<2000;i++)
{
     a[i] = b[a[i]];
     c[i] = d[i]^10;
}
```

```
For (i=0;i<8000;i++)
{
     a[i] = b[a[i]];
     c[i] = d[i]^10;
}
```

```
For (i=0;i<10000;i++)
{
     a[i] = b[a[i]];
     c[i] = d[i]^10;
}
```

DP

Overview of PeaPaw



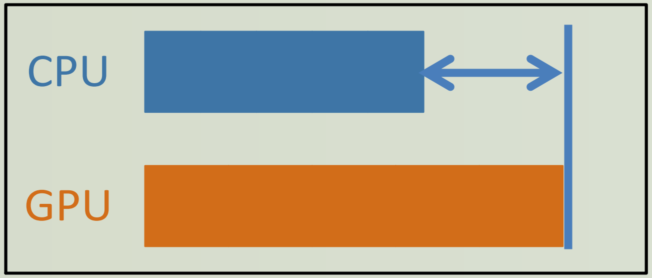
- What parameters should be used?
- How to obtain parameter values?

Modeling Performance/Energy Estimation in PeaPaw

Roofline models of performance & energy

- Basic roofline model of performance [Williams, CACM, 2009]
  - I: operational intensity = flop/byte
  - Gflop/s = min(Peak Gflop/s, Peak Mem BW * I)
  - Simple extension to system level
    - CPU Gflop/s = min(Peak CPU Gflop/s, Peak CPU Mem BW * I_C)
    - GPU Gflop/s = min(Peak GPU Gflop/s, Peak GPU Mem BW * I_G)
    - Ideal CPU+GPU Gflop/s = CPU Gflop/s + GPU Gflop/s
- Other roofline models of performance [Nugteren, CF, 2012][Llic, CAL, 2014]
  - Only consider single processor platforms
- Roofline model of energy[Choi, IPDPS, 2013]
  - $E_{processor} = E_{flops} + E_{bytes} + p_{static} * T_{processor}$

CP-independent



PeaPaw - performance

$$\text{CPU+GPU Gflops/s} = \frac{N_{f-c} + N_{f-g}}{\max(\underbrace{t_{f-c}N_{f-c},\ t_{b-c}N_{b-c}}_{\text{CPU performance}},\ \underbrace{t_{f-g}N_{f-g},\ t_{b-g}N_{b-g}}_{\text{GPU performance}})}$$

CPU cores
- $t_{f\text{-}c}$(time per flop)
- $N_{f\text{-}c}$(# of flops)

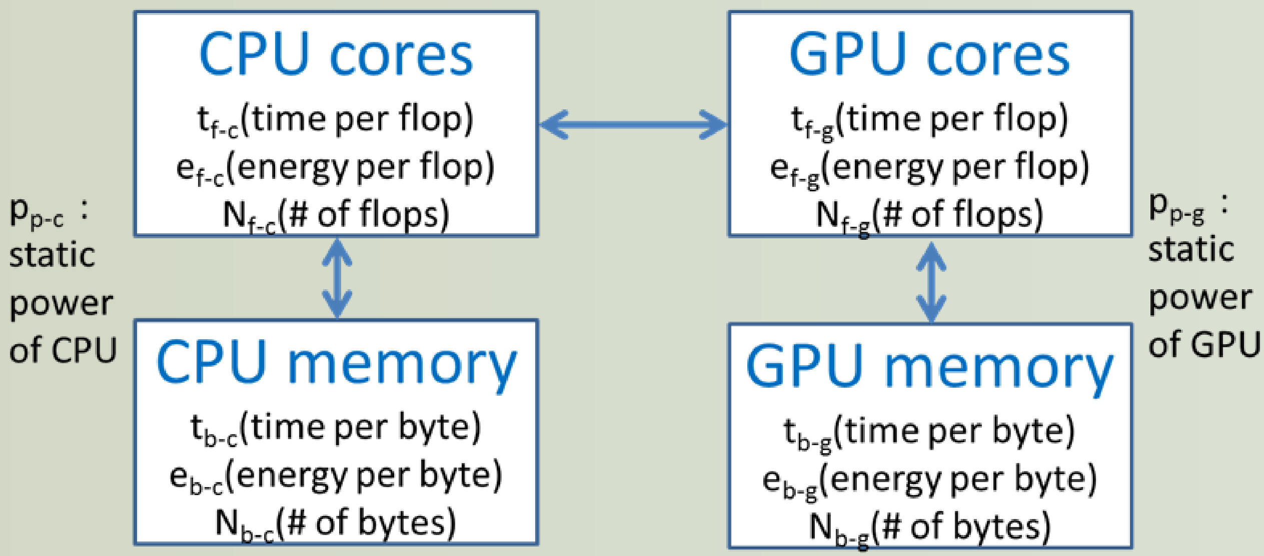
GPU cores
- $t_{f\text{-}g}$(time per flop)
- $N_{f\text{-}g}$(# of flops)

CPU memory
- $t_{b\text{-}c}$(time per byte)
- $N_{b\text{-}c}$(# of bytes)

GPU memory
- $t_{b\text{-}g}$(time per byte)
- $N_{b\text{-}g}$(# of bytes)

PeaPaw – energy

$$E_{total} = \underbrace{e_{f\text{-}c}N_{f\text{-}c}+e_{b\text{-}c}N_{b\text{-}c}}_{\text{CPU dynamic energy}}+\underbrace{e_{f\text{-}g}N_{f\text{-}g}+e_{b\text{-}g}N_{b\text{-}g}}_{\text{GPU dynamic energy}}+\underbrace{(p_{p\text{-}c}+p_{p\text{-}g})*\max(t_{f\text{-}c}N_{f\text{-}c},\ t_{b\text{-}c}N_{b\text{-}c},\ t_{f\text{-}g}N_{f\text{-}g},\ t_{b\text{-}g}N_{b\text{-}g})}_{\text{Total static energy}}$$



Getting parameter values

- WP parameters values: $N_{f\text{-}c}$, $N_{b\text{-}c}$, $N_{f\text{-}g}$, $N_{b\text{-}g}$
  - Code analysis via control-data flow graph
  - Code inspection

- Hardware parameter values
  - $t_{f\text{-}c}$, $t_{f\text{-}g}$: peak performance from specification
  - $t_{b\text{-}c}$, $t_{b\text{-}g}$: realistic peak performance through benchmarks (e.g., STREAM)
  - $e_{f\text{-}c}$, $e_{b\text{-}c}$, $e_{f\text{-}g}$, $e_{b\text{-}g}$, $p_{p\text{-}c}$, $p_{p\text{-}g}$
    - $n_f * e_f + n_b * e_b + p_s * t = E$
    - Run different tests, each test has different n_f and n_b
    - Measure E and t for each test
    - Use linear training to get e_f, e_b and p_s

Essential WP parameters

- Reduce the WP parameters
  - Dataset size independent
  - Easy for data analysis

$$\{N_{f\text{-}c,}\ N_{b\text{-}c,}\ N_{f\text{-}g,}\ N_{b\text{-}g}\} \longrightarrow \{I,\ I_{c,}\ I_g\}$$

$$\text{CPU+GPU Gflops/s} = \frac{N_{f-c} + N_{f-g}}{\max(t_{f-c}N_{f-c},\ t_{b-c}N_{b-c},\ t_{f-g}N_{f-g},\ t_{b-g}N_{b-g})}$$

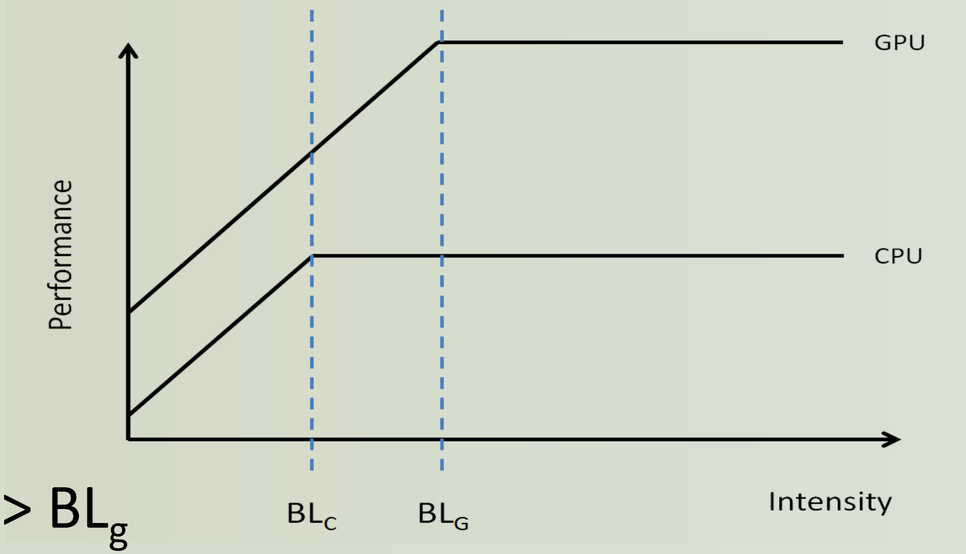
$$\downarrow$$

$$\text{CPU+GPU Gflops/s} = \max\{t_{f\text{-}c}\frac{I_c(I-I_g)}{I(Ic-Ig)},\ t_{b-c}\frac{I-Ig}{I(Ic-Ig)},\ t_{f\text{-}g}\frac{I_g(I-Ic)}{I(Ig-Ic)},\ t_{f\text{-}g}\frac{I-Ic}{I(Ig-Ic)}\}$$

PeaPaw Guidelines

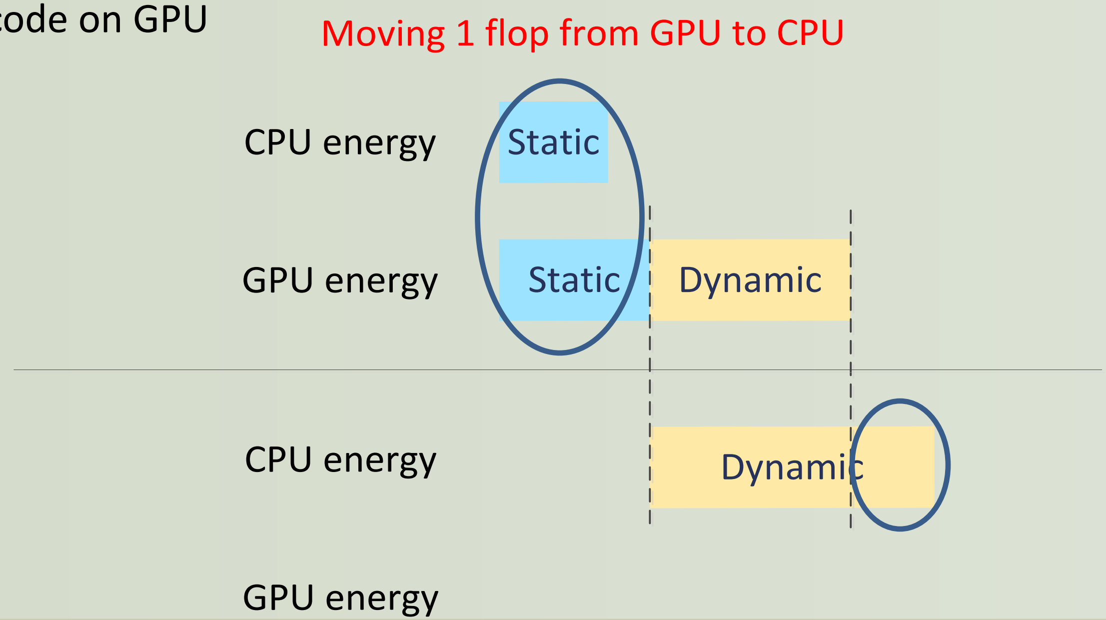
Workload partitioning guidelines - performance

- Machine balance: relative data handling capability
  - $BL_c = t_{b-c} / t_{f-c}$
  - $BL_g = t_{b-g} / t_{f-g}$
- Performance oriented guidelines (P-Paw)
  - CPU_DP-GPU_DP: $BL_c = BL_g$
    - DP on both CPU and GPU
  - CPU_MEM-GPU_COMP: $BL_c > BL_g$
    - Memory intensive part on CPU, compute intensive part on GPU
  - CPU_COMP-GPU_MEM: $BL_c > BL_g$
    - Compute intensive part on CPU, memory intensive part on GPU



Workload partitioning guidelines - energy

- Gradient energy: dynamic and static energy tradeoff
  - $\Delta GP_f = |e_{f\text{-}c} - e_{f\text{-}g}| - (P_{s\text{-}C} + P_{s\text{-}G})^* t_{f\text{-}g}$
  - $\Delta GP_b = |e_{b\text{-}c} - e_{b\text{-}g}| - (P_{s\text{-}C} + P_{s\text{-}G})^* t_{b\text{-}g}$
- Energy-oriented guidelines (E-Paw)
  - GPU-only: $\Delta GP_f > 0$, $\Delta GP_g > 0$, $e_{f\text{-}c} > e_{f\text{-}g}$ and $e_{b\text{-}c} > e_{b\text{-}g}$
    - Use GPU for the whole workload
  - CPU_MEM-GPU_COMP: $\Delta GP_f > 0$, $\Delta GP_g > 0$, $e_{f\text{-}c} < e_{f\text{-}g}$ and $e_{b\text{-}c} > e_{b\text{-}g}$
    - Memory intensive code on CPU and computation intensive code on GPU
  - Race-to-halt: $\Delta GP_f + \Delta GP_f < 0$
    - Follow the P-Paw to achieve the highest performance



Experimental Evaluation

Processors

Processors	Types	Architectures	# of cores	Frequency (GHz)
Intel i7 2600K	CPU	Sandybridge	4	3.4
Intel i3 2100T	CPU	Sandybridge	2	2.5
NVIDIA GTX Titan	GPU	Kepler	2688	0.84
NVIDIA GTX 750	GPU	Kepler	750	0.51

Processors	T_f (pS)	T_b (pS)	E_f (pJ)	E_b (pJ)	P_s (W)
Intel i7 2600K	9.5	65.9	118	462	26.8
Intel i3 2100T	25	73	135	581	9.7
NVIDIA GTX Titan	0.4	4.2	57	187	64.1
NVIDIA GTX 750	1.9	14.8	78	169	16.4

Guidelines – performance & energy

Platforms	BL_C	BL_G	P-Paw category
i7+Titan	0.9	10.3	CPU_MEM-GPU_COMP
i7+750	0.9	7.8	CPU_MEM-GPU_COMP
i3+Titan	0.4	10.3	CPU_MEM-GPU_COMP
i3+750	0.4	7.8	CPU_MEM-GPU_COMP

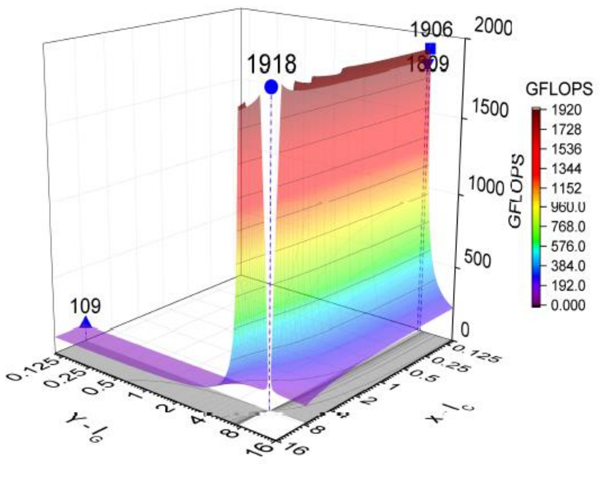
Platforms	ΔGP_f	ΔGP_b	$P_{s\text{-}C} + P_{s\text{-}G}$	E-Paw category
i7+Titan	9.5	65.9	118	CPU_MEM-GPU_COMP
i7+750	25	73	135	Race-to-halt
i3+Titan	0.4	4.2	57	GPU-only
i3+750	1.9	14.8	78	GPU-only

Workload partitions (WPs)

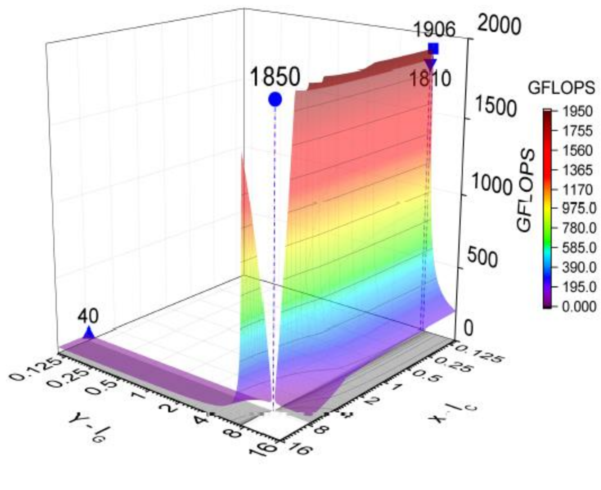
Workloads	CPU-only	GPU-only	p-DP	CP
SA	{7.6, 7.6, 0}	{7.6, 0, 7.6}	{7.6, 7.6, 7.6}	{7.6, 0.1, 8}
LA	{0.24, 0.24, 0}	{0.24, 0, 0.24}	{0.24, 0.24, 0.24}	{0.24, 0, 0.25}
DA	{4.4, 4.4, 0}	{4.4, 0, 4.4}	{4.4, 4.4, 4.4}	{4.4, 1.1, 4.9}

- Synthetic application (SA)
- Linear algebra (LA)
- Data assembly (DA)

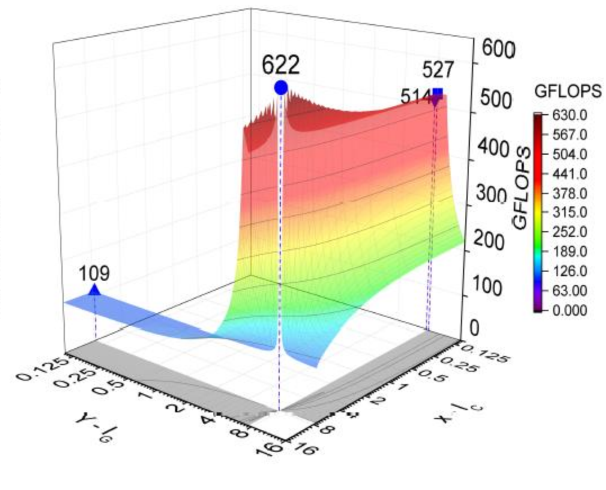
Validation – performance of SA



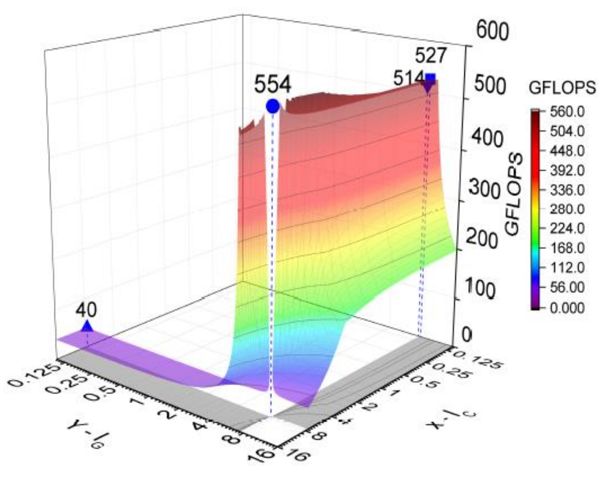
(a) i7+Titan



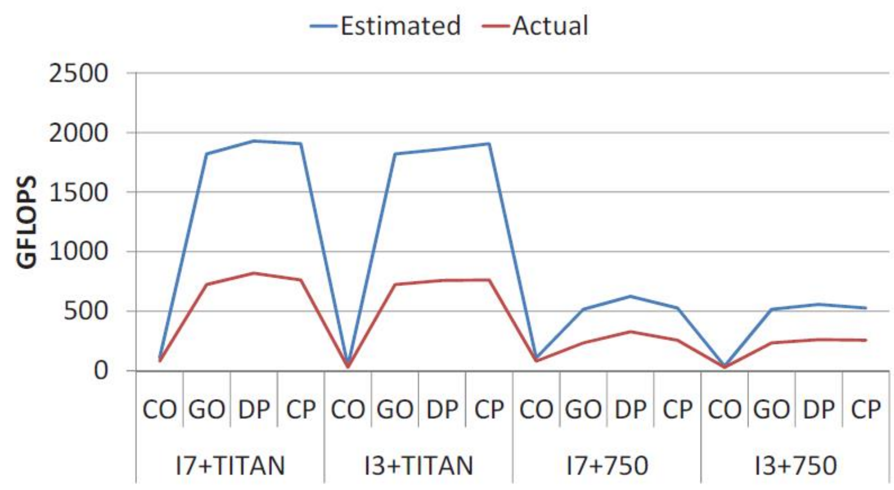
(b) i3+Titan



(c) i7+750

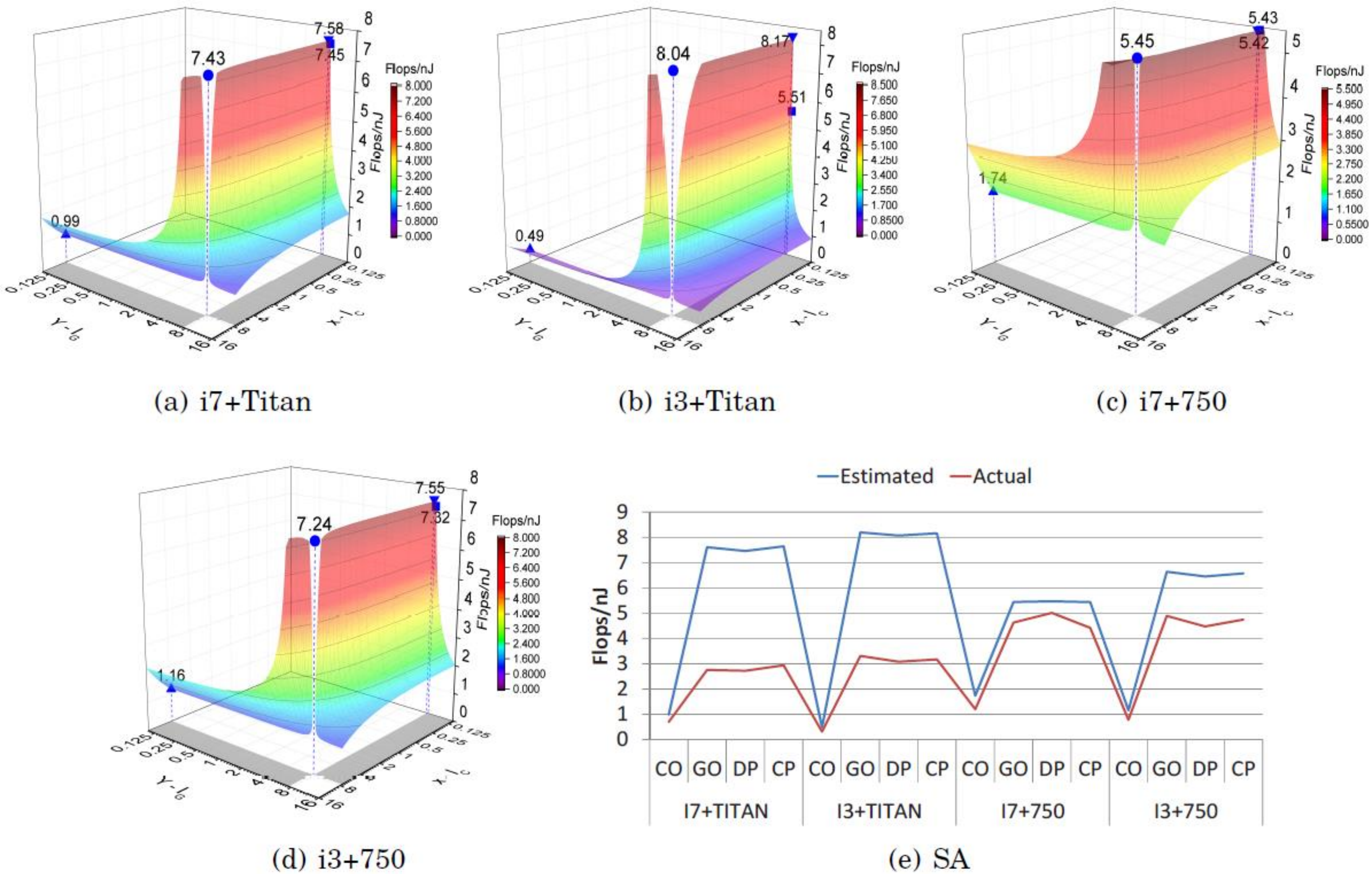


(d) i3+750



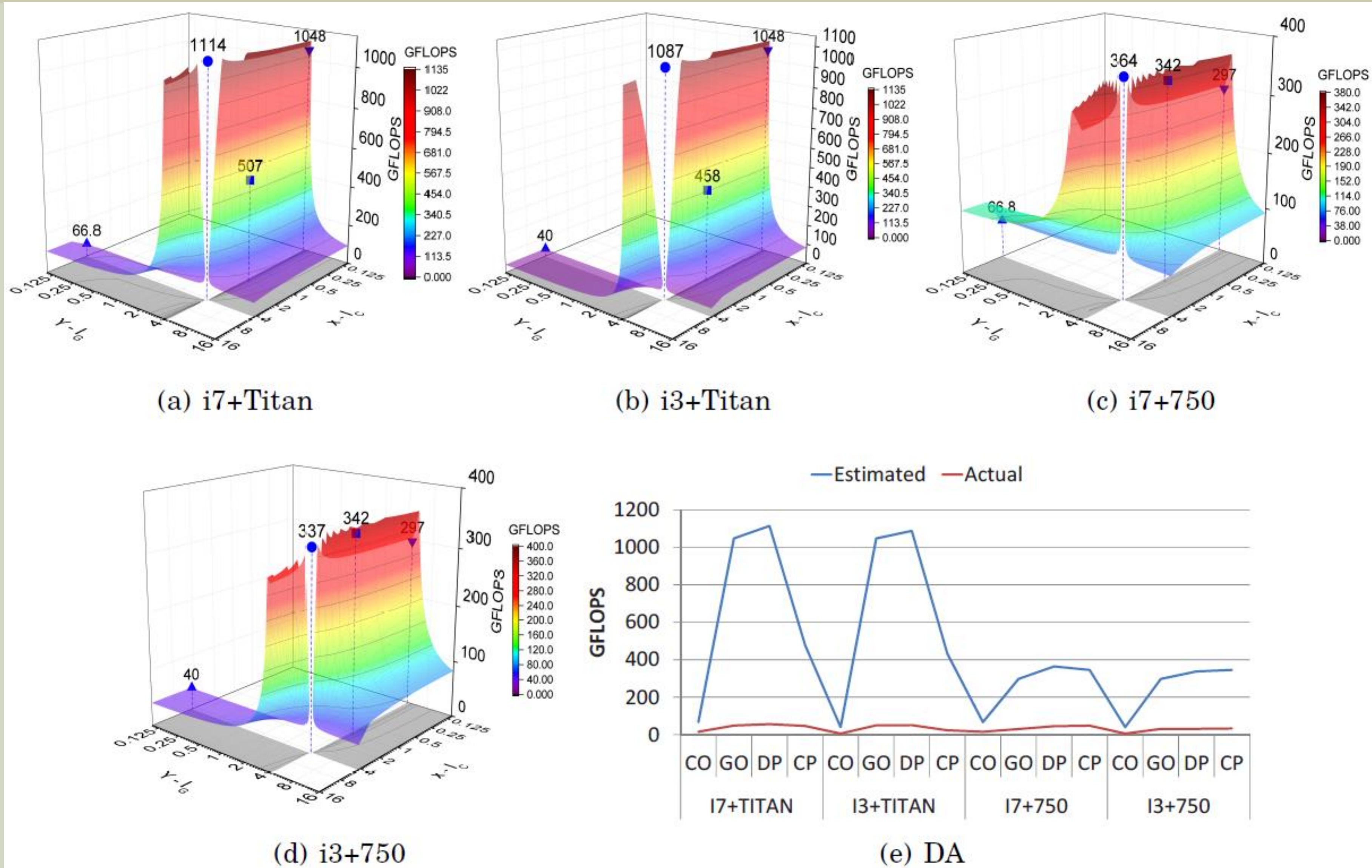
(e) SA

Validation – energy efficiency of SA



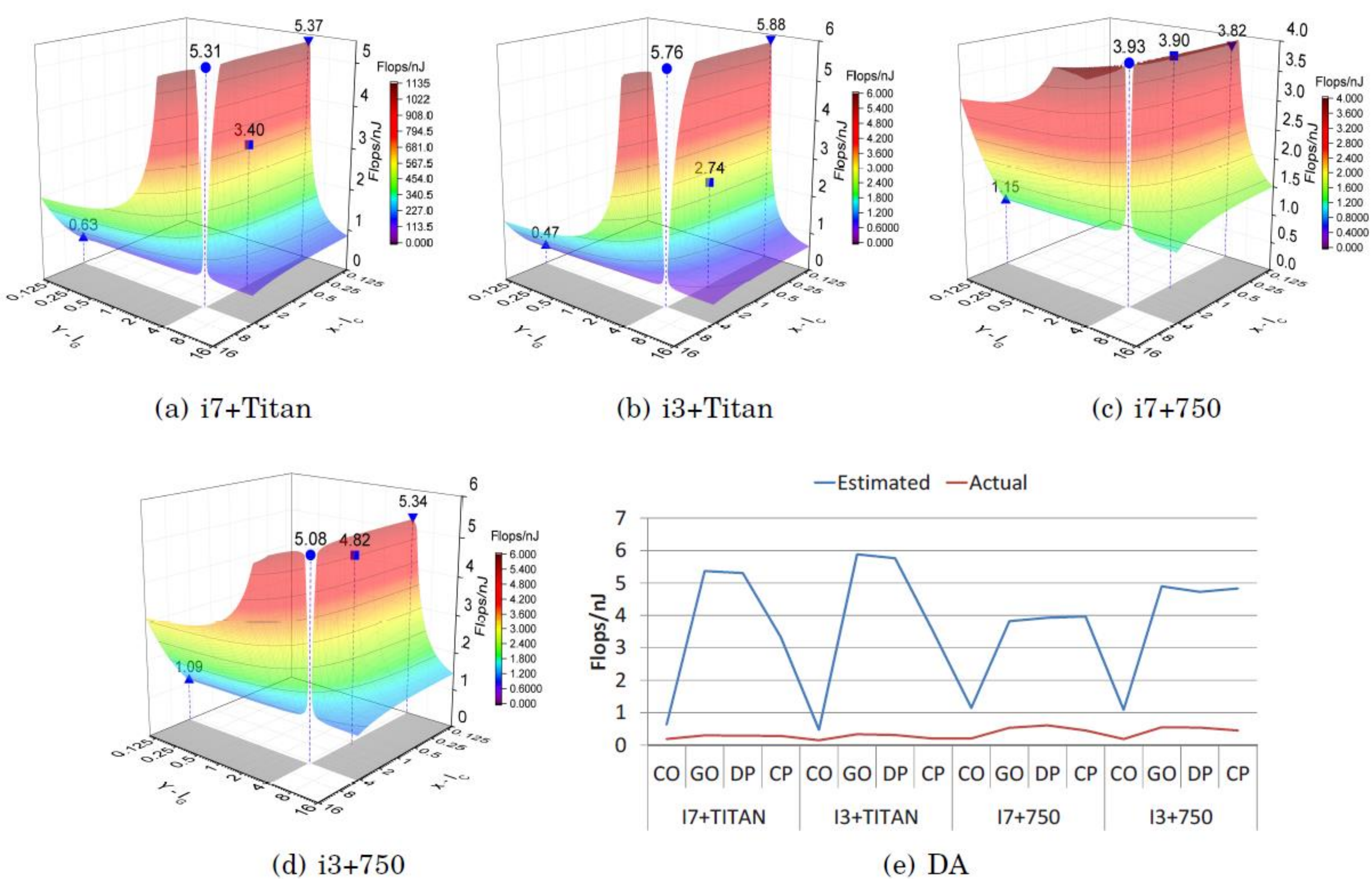
(a) i7+Titan (b) i3+Titan (c) i7+750 (d) i3+750 (e) SA

Validation – performance of DA



(a) i7+Titan (b) i3+Titan (c) i7+750 (d) i3+750 (e) DA

Validation – energy efficiency of DA



(a) i7+Titan (b) i3+Titan (c) i7+750

(d) i3+750 (e) DA

Summary and future work

- The PeaPaw framework
  - Performance/energy modeling of heterogeneous platforms
  - Performance oriented and energy oriented guidelines for workload partitioning
  - P-Paw and E-Paw categories for guideline selection
- Future work
  - Improve the guideline quality
  - Improve the accuracy of performance/energy estimation
  - Add support to integrated CPU+GPU platforms

Thank you!