Integrated Matrix Extension (IME)

Task Group Meeting

Guido Araujo
Jose Moreira

04/22/24

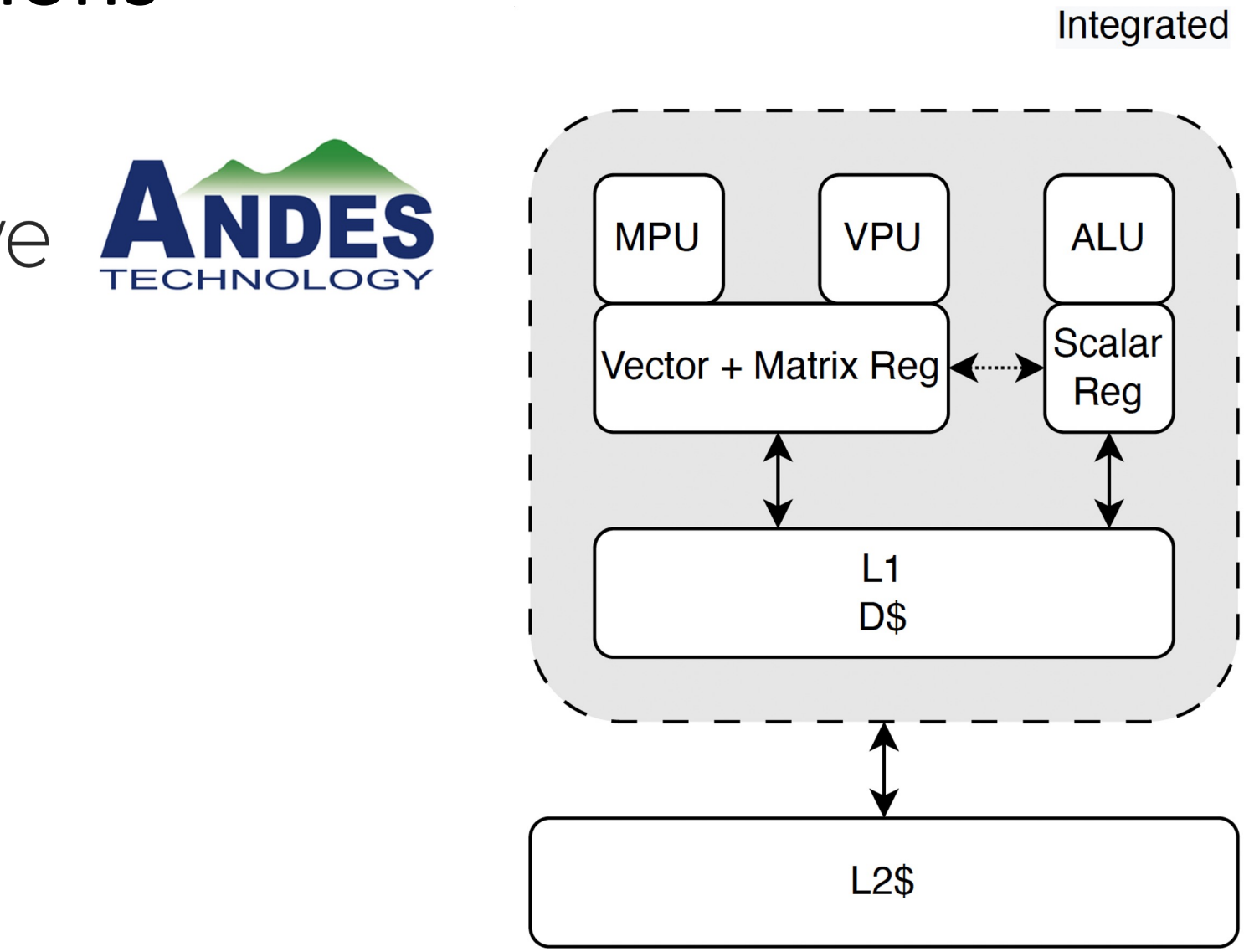
Agenda

- Revisiting what we have achieved so far
- Definition of gaps, agenda and working groups

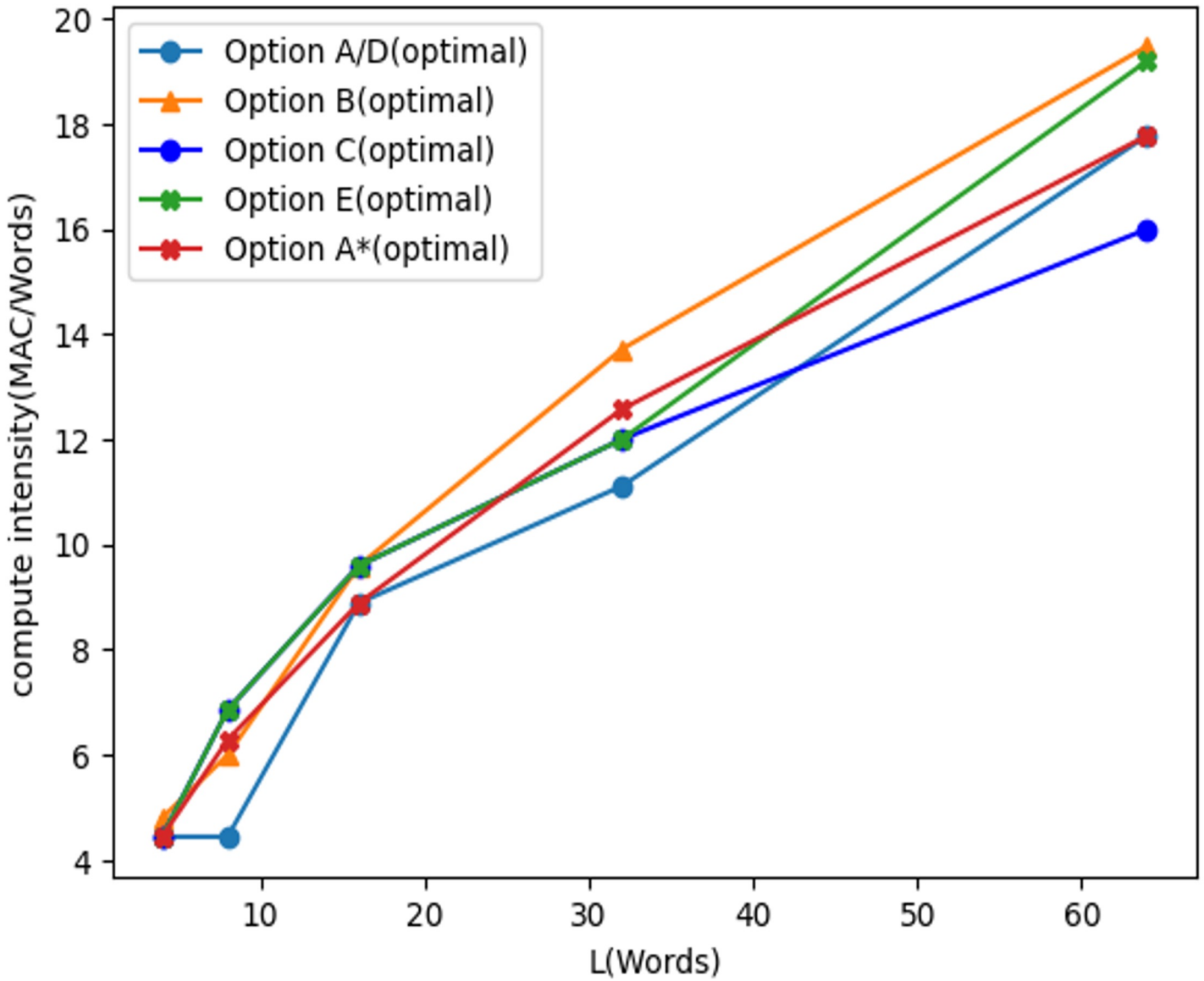
What we have achieved so far

- Many presentations
- ISA modeling and analytical evaluation
  - IBM and Esperanto: Architectural options
- Technical discussions on the merits of each approach
- Analytical estimates for metrics
  - Andes and Unicamp: Computational intensity evaluation
  - Alibaba and Unicamp: Preliminary workload evaluation
  - Andes and Ventana: Register port analysis (uArch)
  - Unicamp: Packing cost evaluation*
  - Unicamp: POWER10 vector-matrix transfers*

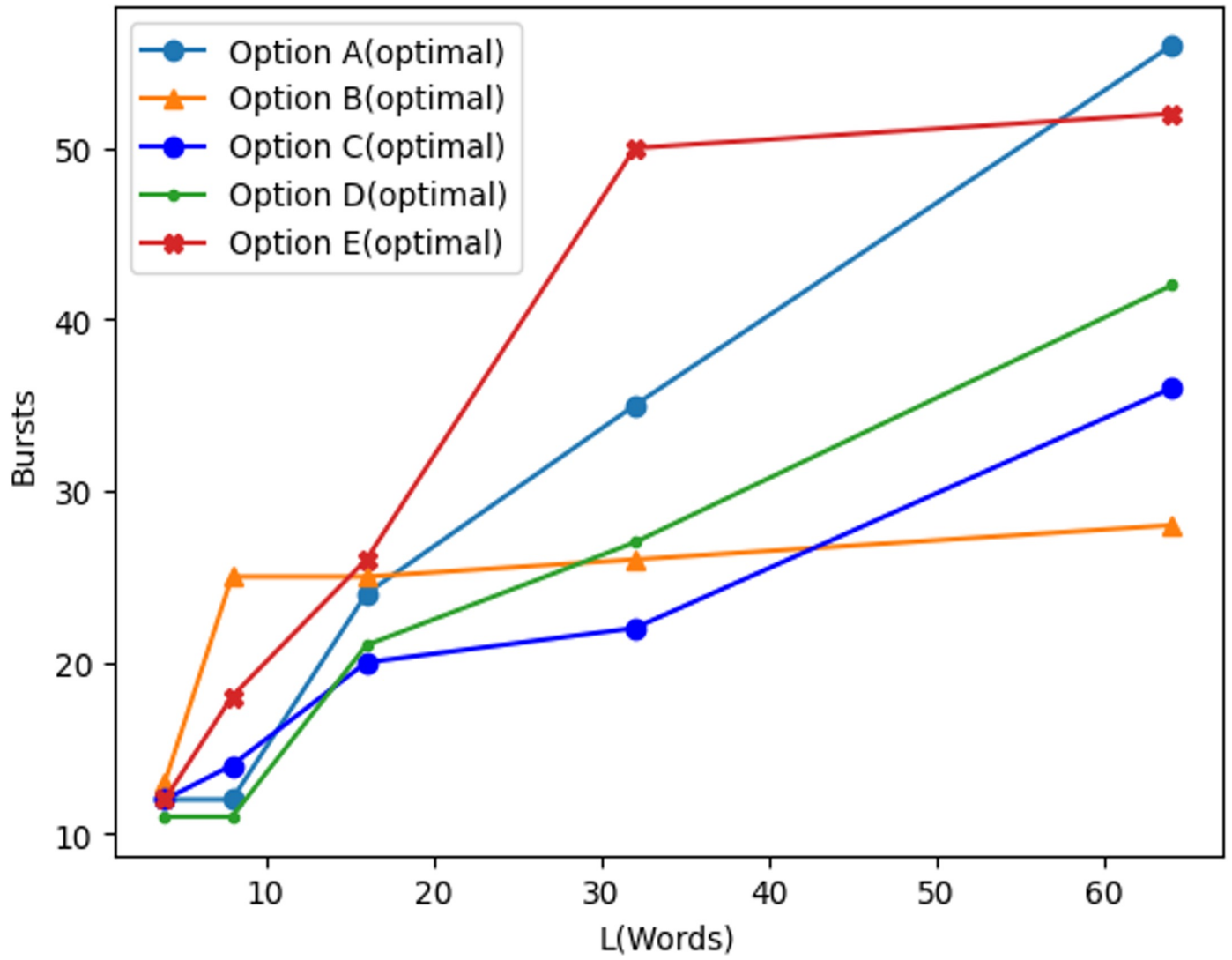
Many presentations



Computational Intensity



Evaluating impact on packing*



Charter Criteria Discussion

- Based on proposing Matrix-TG → 9 explicitly guide-lines
- Suggest to add Metrics for Performance/uArch Cost

No.	Guides	Option A	Option B	Option C	Option D	Option E
1	**VLEN agnostic** at binary level	Not Disclosed while L!=λ^2	To Be Discussed (seems support?)		Not Disclosed while L!=λ^2	(1) Source Compliant (2) Binary Compliant for vmul/fused ISA
2	**Deterministic Result** (FMAC rounding/ordering)	Shall support	Shall support	Shall support	Shall support	Support if finalized
3	**Re-producible result** with plain scalar/vector	Shall support	Shall support	Shall support	Shall support	Support (BIT TRUE test)
4	**Near peak (~90%) performance** for GEMM kernels is possible					GEMM kernels Near Peak U-rate
5	**Higher (~2X) performance** than vector					Over (>3X) enhancement than RVV
6	Maximization **computation intensity** for GEMM kernels					GEMM kernels Near Peak U-rate
7	Minimization additional **architecture state**	None new state	None new state	None new state	Not Support (New Streaming buffer for A/B)	None new state (not considering ZOB)
8	**Live-migration** with larger vector registers	Not Support for different lambda	To Be Discussed (seems support?)	Not Support for different vector element type	Not Support for different lambda	Under Working (AMM 2.0)
9	Proper Support for **packing/reformat data**	May Need Additional handling while L!=λ^2	Support (No additional interleaving/shuffle required)	Support (No additional interleaving/shuffle required)	May Need Additional handling while L!=λ^2	Support (No additional interleaving/shuffle required)
10	**Metrics for Performance/uArch cost (Suggest to consider)**	Feasible uArch cost(VRF R/W) for Specific MAC Performance	High uArch cost(VRF R/W) for MAC Performance†	feasible uArch cost(VRF R/W) for MAC Performance	Feasible uArch cost(VRF R/W) for MAC Performance	Feasible uArch cost(VRF R/W) for MAC Performance

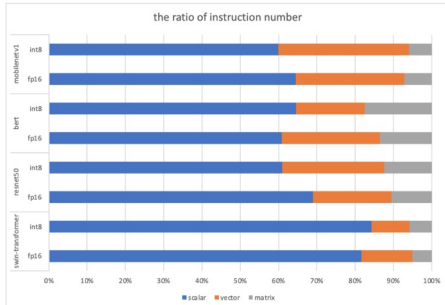
†: refer details at

Workloads and Benchmarking

- [Alibaba] Analysis of matrix and vector instruction distribution in ML Models

scalar vs vector vs matrix

Statistics here are the distribution of the number of instructions, not the distribution of computing power.



networks	swin-transformer		resnet50		bert		mobilenetv1	
data types	fp16	int8	fp16	int8	fp16	int8	fp16	int8
scalar	81.64%	84.32%	68.94%	60.95%	60.84%	64.53%	64.49%	59.91%
vector	13.29%	9.82%	20.54%	26.59%	25.66%	17.96%	28.42%	34.14%
matrix	5.07%	5.86%	10.52%	12.47%	13.50%	17.50%	7.09%	5.96%

- [Unicamp] Analysis of vector and matrix transfers in Polybench (HPC benchmark)

	Name	Algorithm	vector = vector op vector		matrix = vector op matrix		vector = vector op matrix		matrix = matrix op matrix	
			lines	%time	lines	%time	lines	%time	lines	%time
3	2mm	D := alpha*A*B*C + beta*D							2 mac: (83-89) (90-96) with scalar	
4	3mm	G := (A*B)*(C*D)							3 mac : (79-85) (87-93) (95-101)	
5	atax	y := (x*A')*A + y					2 mac: (68-72) (68-75)			
6	bicg	s := r*A + s q := A*p + q					2 mac: (77-85) (77-85)			
7	cholesky	Algorithm to find L, in which A := LL' (triangular application)	2mac: (70-71) (76-77)							
8	doitgen	R matrix multiplications in cubic buffers							1 macs: (69-74) 1 movs: (75-76)	
9	gemm	C := alpha*A*B + beta*C							1 mac (77-83) with scalar	
10	gemver	A := A + u1.v1 + u2.v2 x := beta*A*y +z w := alpha*A*x	1 add: (100-101)		2 mac: (92-94)		1 mac: (96-98) with scalar 1 mac: (103-105) with scalar		2 add (92-94)	
11	gesummv	y := alpha*A*x + beta*B*x			2 macs (76-84) (76-84) with scalar					
12	mvt	x1 := x1 + A*y1 x2 := x2 + A*y2					2 mac: (75-77) (79-81)			
13	symm	C := alpha*A*B + beta*C					1 mac: (81-84) with scalar			
14	syr2k	C := alpha*A*B' + alpha*B*A' + beta*C							2mac: (79-85) with scalar	
15	syrk	C := alpha*A*A' + beta*C							1mac: (75-78) with scalar	

Agenda

- Revisiting what we have achieved so far
- Definition of gaps, agenda and working groups

Gaps and agenda (14 meetings)

- Today: Definition of working groups
- 2 Meetings: Consolidation of the feature table
- 1 Meetings: Definition of workloads and benchmarking
- 2 Meetings: Discussion and definition of two candidates
- 3 Meetings: Quantitative analysis
- 2 Meetings: Definition of the final candidate
- 4 Meetings: Discussion and writing of the specification

Working groups

Groups	Coordinator	Members
Option A and A*		
Option B		
Option C		
Option D		
Option E		
Workloads and Benchmarking	Guido	

Quantitative Approach

Polybench
Convbench
Other

Options'
Simulators

Metrics

- Option A
- Option B
- Option C
- Option D
- Option E