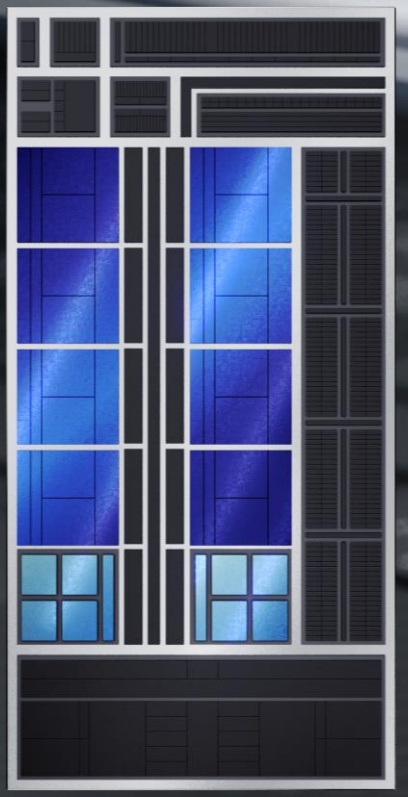
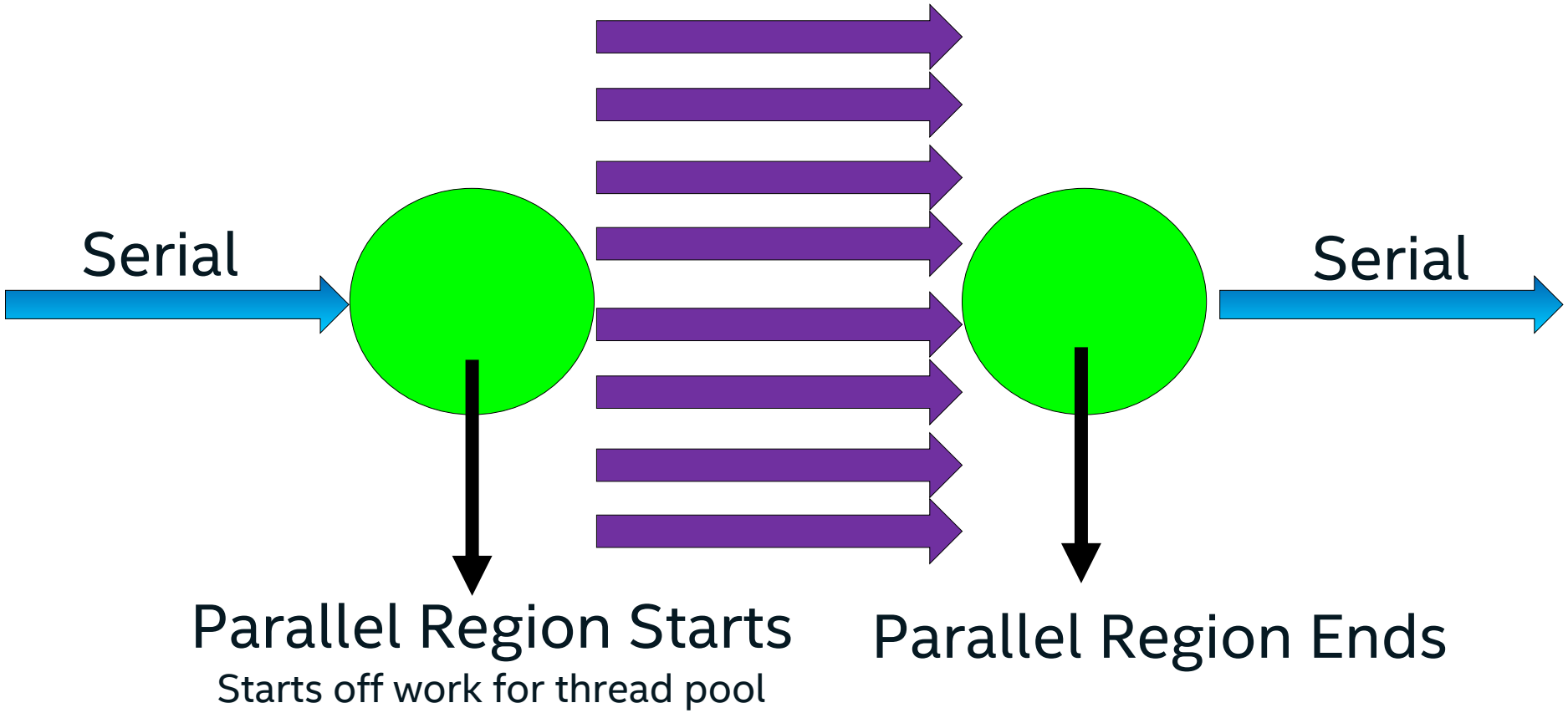
Scalable Tools Workshop 2022:
Using Hybrid Cores to Optimize Performance, Power and Throughput

Michael Chynoweth – Intel Fellow
Contributors: Rajshree Chabukswar, Yuli Mandelblat, Sneha Gohad, Nikhil Rukmabhatla, Ahmad Yasin, Wei Xiao, Eli Hernandez, Patrick Konsor, Joe Olivas

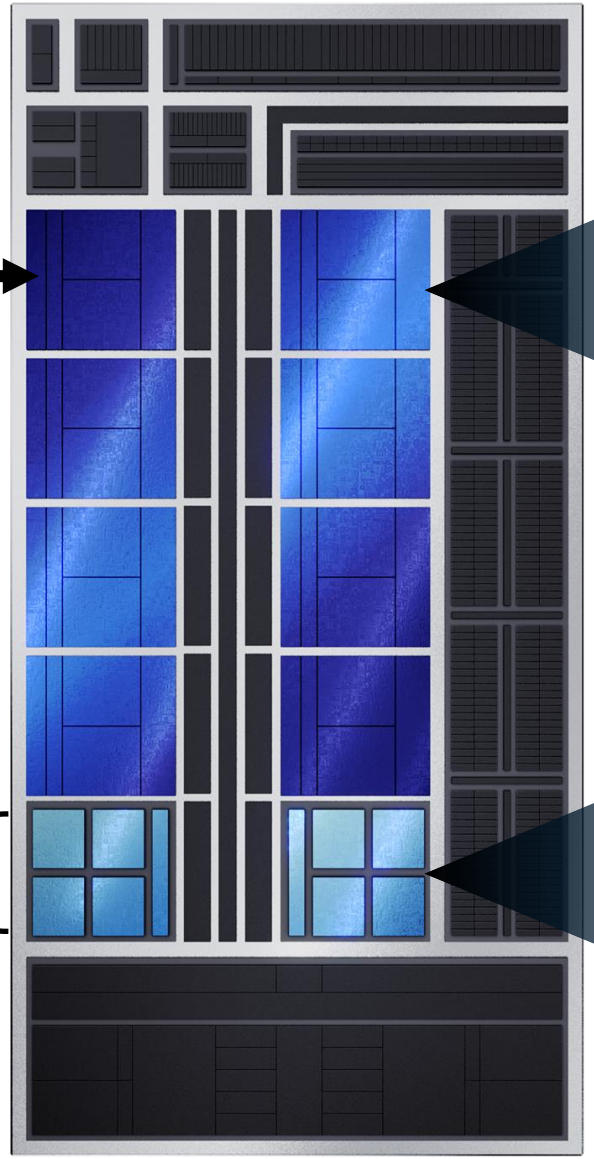


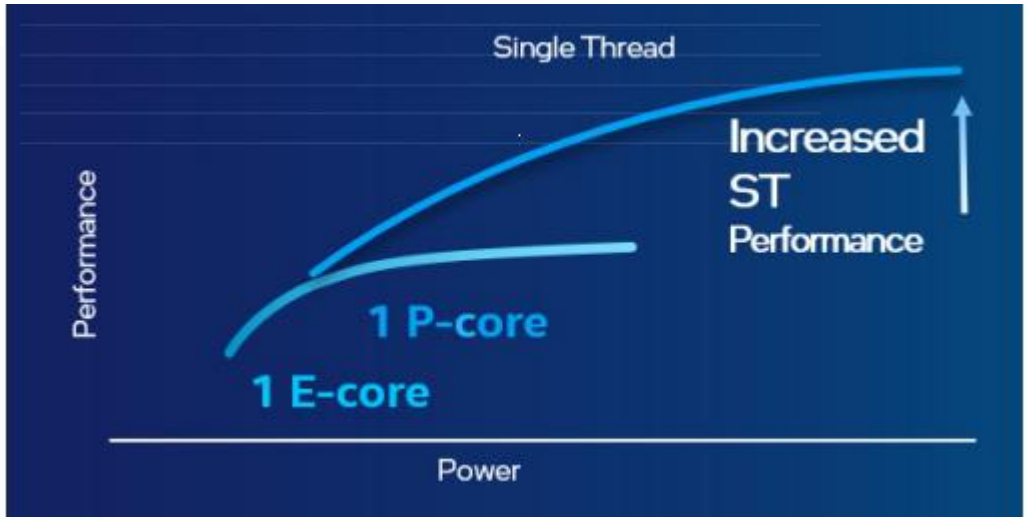
intel®

Example of Challenge in Multi-Threaded Scaling

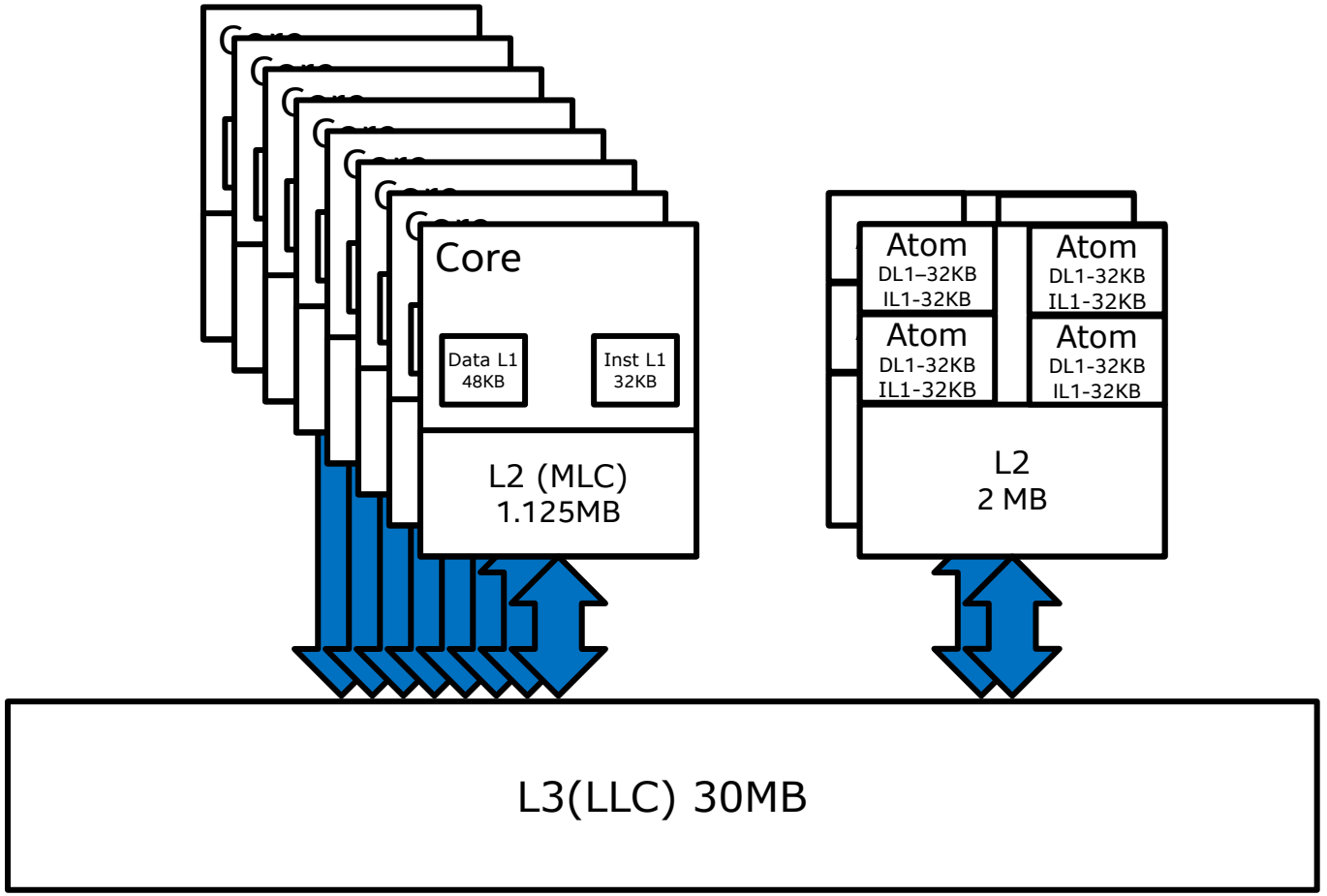


DESIGN GOALS FOR HYBRID (GOING BACK YEARS)





Intel® Core™ i9-12900K

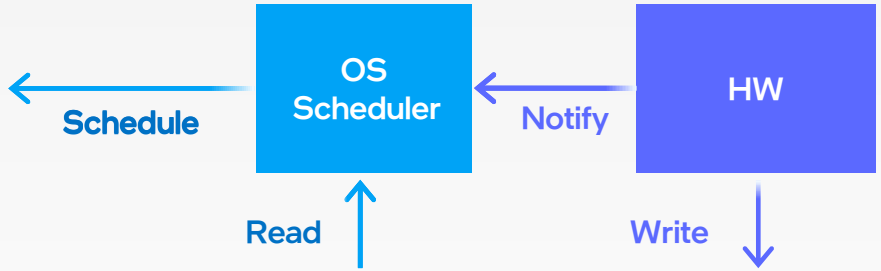


- P-Core:
  - Data L1 – 48KB;
  - Code L1 – 32KB;
  - L2 (MLC) – 1.125MB*
- E-Core:
  - Data L1 – 32KB;
  - Code L1 – 32KB;
  - L2 – 2 MB*
- Shared L3 (LLC) 30MB

* Depends on product SKU definition

Intel **Thread Director -** Architecture

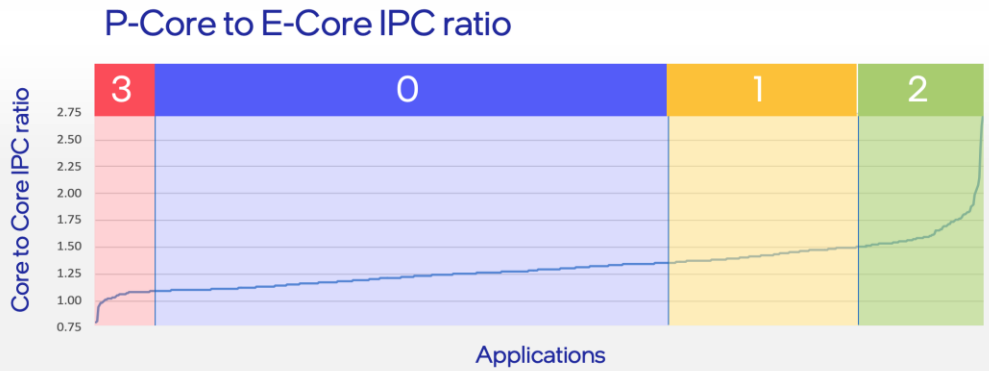
- HW periodically writes a feedback table (EHFI)
  - Function of aggregated load and physics
- OS scheduler selects the best core allocation for the SW thread runtime properties and class
  - Most performing core -or-
  - Most energy efficient core
  - Communicates Energy Performance Preference



Logical process

Thread Director Table in Main Memory

Core ID	Class3		Class 2		Class 1		Class 0	
	EE	Perf	EE	Perf	EE	Perf	EE	Perf
0	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap
1	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap
2	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap
	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap
LPn-1	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap	EE Cap	Perf Cap

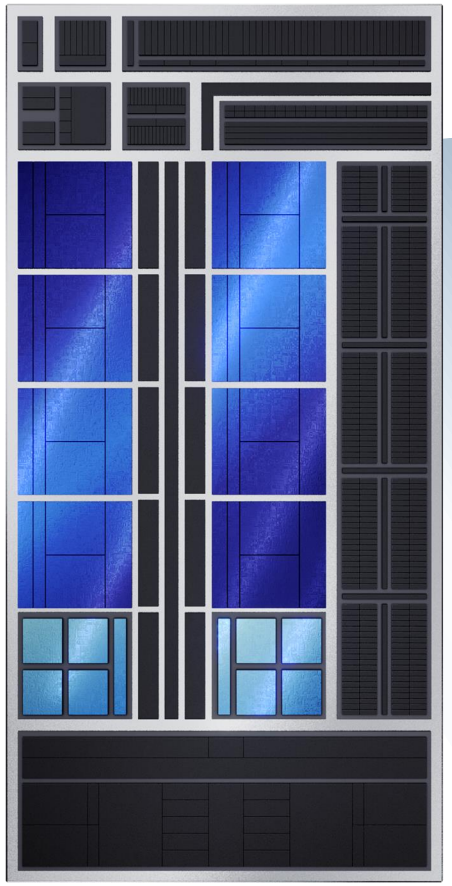


IPC-to-IPC classes

In some scenarios:
P-core may be more efficient
E-core may be higher performance
0 → Hint do not schedule

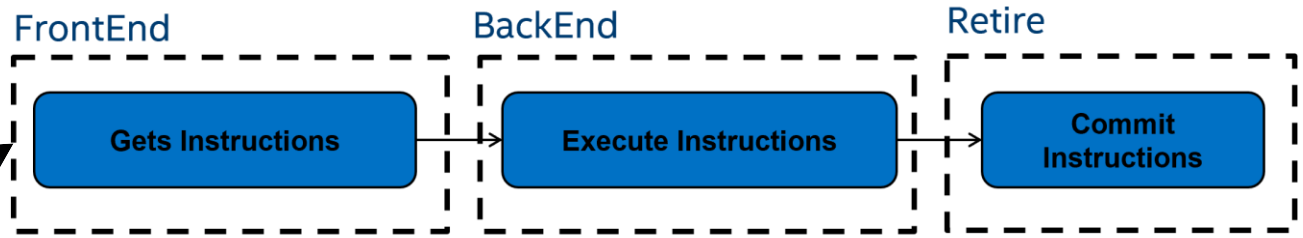
Energy Eff.
Performance

Hybrid Core Pipelines Differ:
Converging Top Down Metrics on Both P-Core and E-Core



Simplify

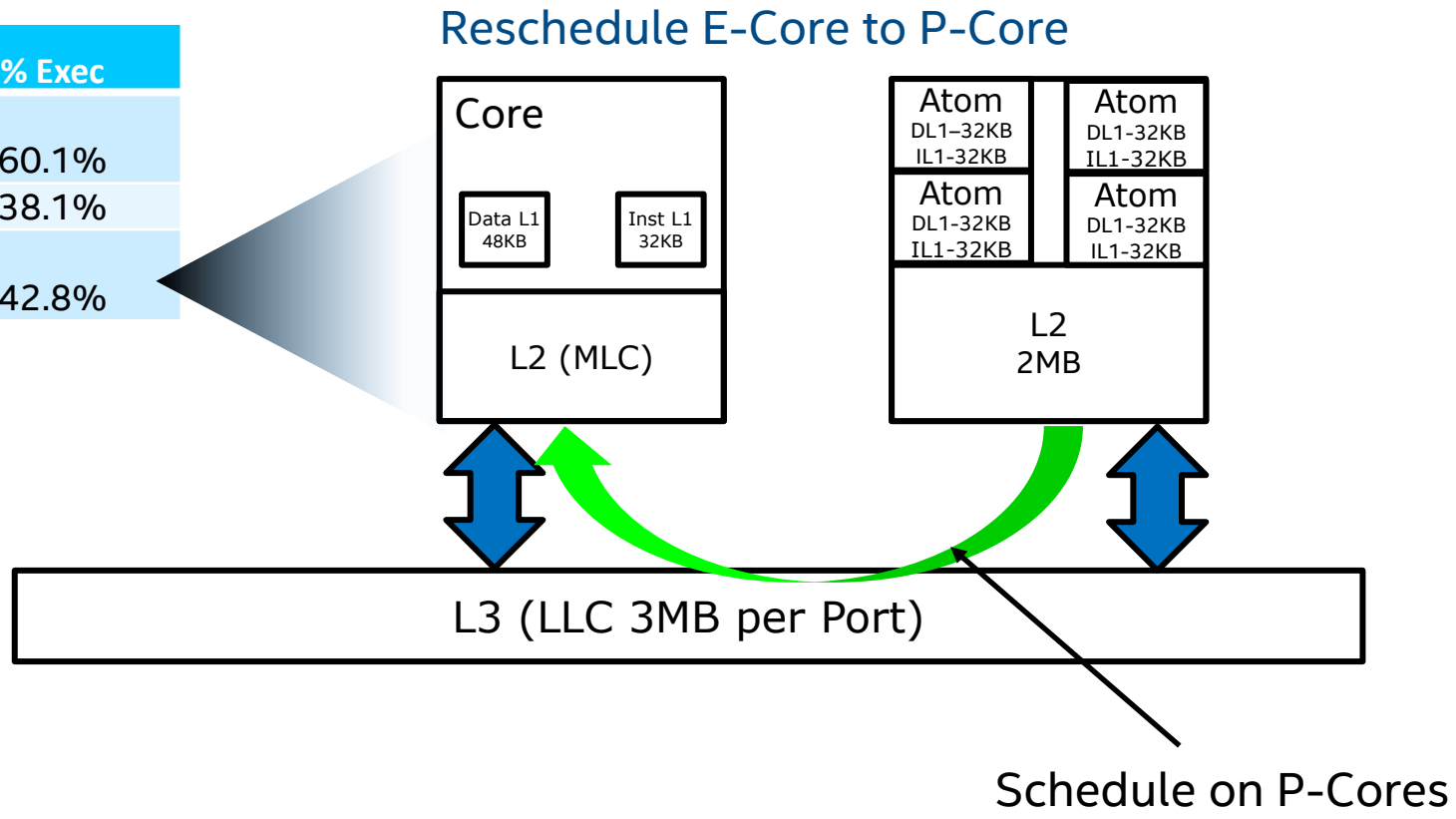
Top Down breaks the pipeline into 4 categories



- Front End Bound = Bound in Instruction Fetch/Decode
- Back End Bound = Usually load latency or execute
- Bad Speculation = When pipeline incorrectly predicts execution
- Retiring = Pipeline is retiring uops

Hardware Guided Scheduling Detecting Execution

Metric	% Exec
Backend Bound	60.1%
Core Bound	38.1%
%AVX_VNNI	42.8%

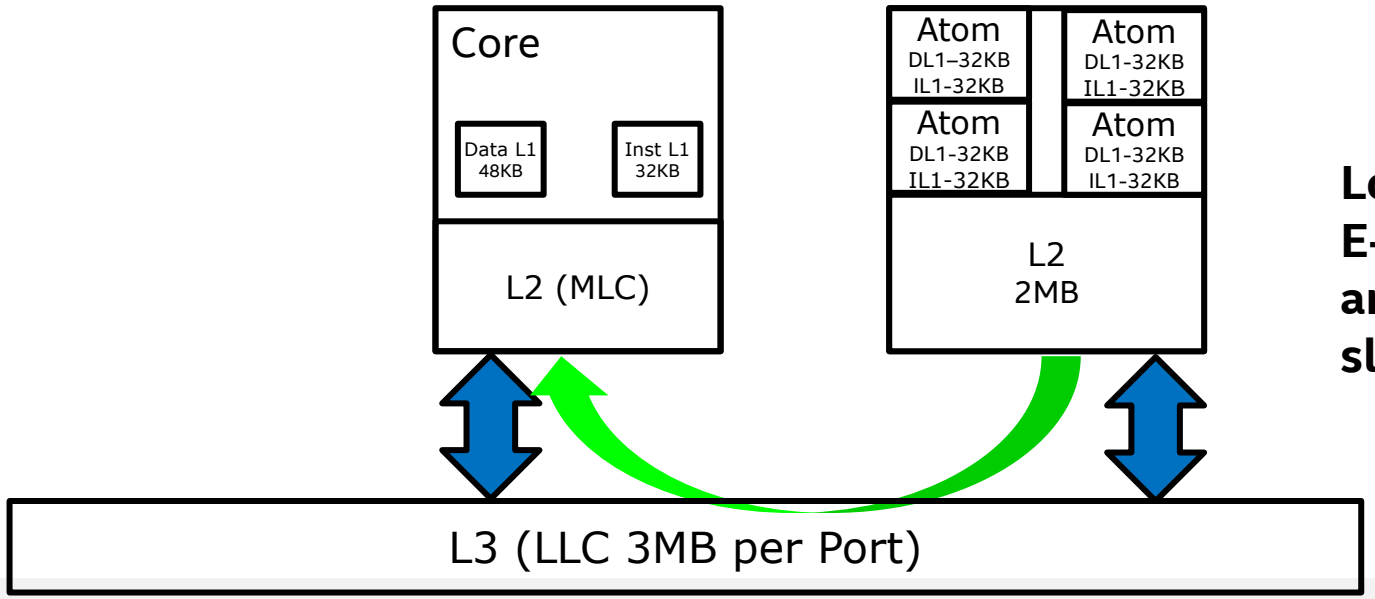


Vector Neural Network Instructions Throughput Gets Moved to P-Core

Example of Issue Resolved with Hybrid Cores: Balance Moving Cores and Migrations

Metric	Non-Hybrid	Hybrid	Non-Hybrid/Hybrid
Frontend_Bound(%)	7.0	5.6	-1.4
Bad_Speculation(%)	6.3	6.9	0.6
Backend_bound(%)	63.8	67.1	3.3
Retiring(%)	22.9	20.4	-2.5

Metric	Non-Hybrid	Hybrid	Non-Hybrid/Hybrid
Backend bound(%)	63.8	67.1	3.3
..Load/Store_Bound(%)	52.8	57.8	5.0
....L3_Bound(%)	17.9	24.1	**6.2**



Losing 6% if we migrate from E-Core -> P-Core too aggressively and threads have small time slice left

Utilizing Top Down to Compare E-Cores and P-Cores

Metric	E-Cores	P-Cores	P-E
Frontend_Bound(%)	28.3	27.2	-1.1
Bad_Speculation(%)	16.7	15.0	-1.6
Backend_Bound(%)	24.7	12.0	-12.7
Retiring(%)	26.6	45.7	19.1

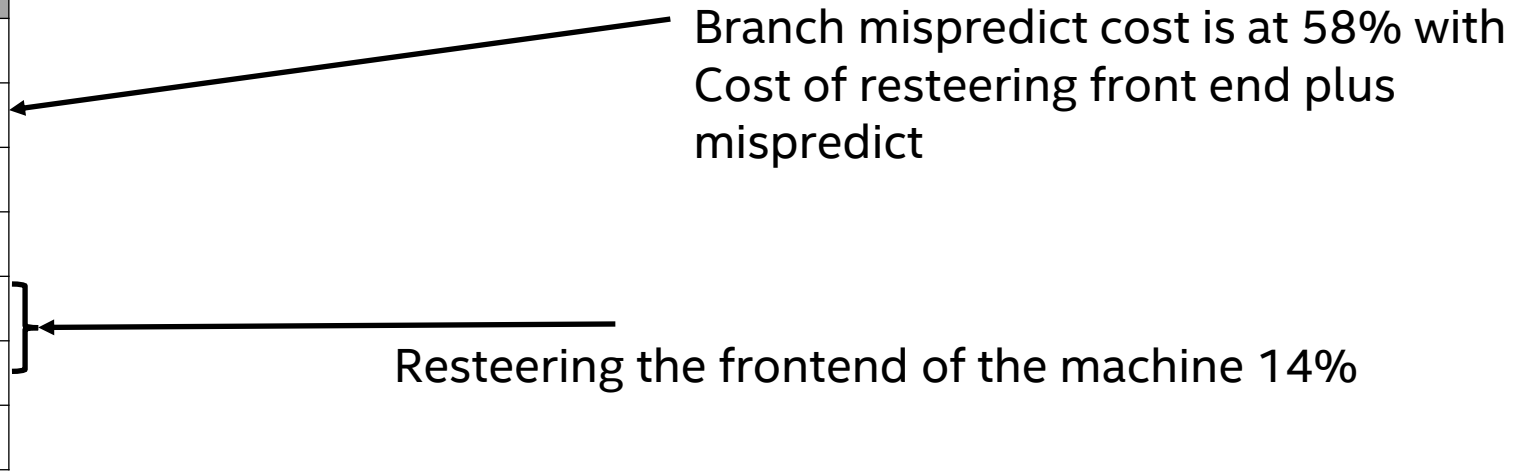
Branch Stat	E-Cores	P-Cores	P-E
..Branch_Mispredicts(%)	16.3	13.1	-3.3
Mispredicts on Conditional	73.6%	86.9%	13.3%
Mispredicts on Indirects	26.0%	13.2%	**-12.9%**

Branch Stat	E-Cores	P-Cores	P/E
BranchMispredict Ratio	0.023	0.019	0.83
Indirect Branch Mispredict Ratio	0.088	0.037	**0.42**

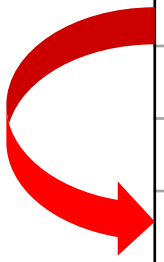
Indirects causing branch prediction problems

Codec Mispredict Cost

Metric	RED8K HTON
Bad_Speculation(%)	44.75
..Branch_Mispredicts(%)	43.58
Frontend_Bound(%)	20.57
..Fetch_Latency(%)	10.95
....Branch_Resteers(%)	13.89
......Mispredicts_Resteers(%)	13.45
..Fetch_Bandwidth(%)	9.62



Branch Both P-Core + E-Core Mispredict

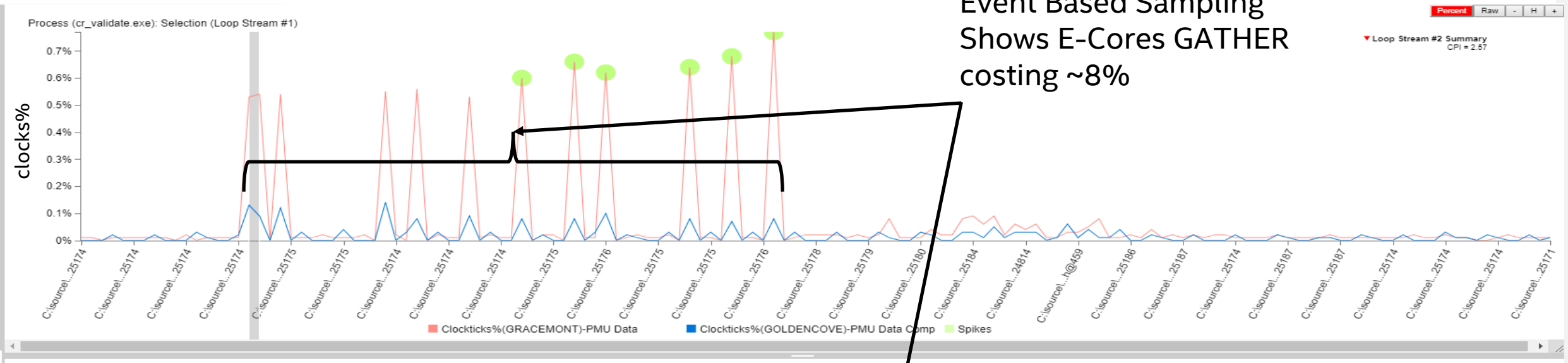


ASM	Notes
1800a9860: FFC9 dec ecx	
1800a9862: 8B02 mov eax,dword ptr [rdx]	
1800a9864: 85C0 test eax,eax	
1800a9866: 7906 jns 1800a986e	7% of mispredicts
1800a9868: 0FBAF01F btr eax,1f	Jump over instruction
1800a986c: F7D8 neg eax	Jump over instruction
1800a986e: 660F6EC0 movd xmm0,eax	
1800a9872: 0F5BC0 cvtdq2ps xmm0,xmm0	
1800a9875: F30F59C1 mulss xmm0,xmm1	
1800a9879: F30F1107 movss dword ptr [rdi],xmm0	
1800a987d: 4883C204 add rdx,4	
1800a9881: 4903F8 add rdi,r8	
1800a9884: 85C9 test ecx,ecx	
1800a9886: 75D8 jnz 1800a9860	

Top Mispredict Fixed with Conditional Moves + Vectorization

Balancing Code Generation Between P-Cores and E-Cores

Extended Processor
Event Based Sampling
Shows E-Cores GATHER
costing ~8%



Selection Granularity

Copy Export

Heirarchy							
HB#	BB#	LN#	Disasm	L	HitCount	Clockticks%(GRACEMONT)	Clockticks%(GOLDENCOVE)
3	3	144	lea r12, ptr [0]	1	348638820	0.01%	0.02%
3	3	145	vgatherqps xmm7, xmmword ptr [r12+ymm4*1], xmm9		348638820	0.53%	0.13%
3	3	146	vgatherqps xmm10, xmmword ptr [r12+ymm11*1], xmm13		348638820	0.54%	0.09%
3	3	147	vmovdqa xmm9, xmm5		348638820	0.00%	0.00%
3	3	148	vgatherqps xmm0, xmmword ptr [r12+ymm12*1], xmm14		348638820	0.54%	0.12%

Instruction	P-Core/GLC %Execution	E-Core/GRT %Execution	GLC-GRT
Gather QWORD Indices (VGATHERQPS) ARG_TYPE: {REG128(XMM), MEM128, REG128(XMM)}	1.1	7.2	**-6.1**
Gather Instruction DWORD Indices (VGATHERDPS) ARG_TYPE: {REG256(YMM), MEM256, REG256(YMM)}	0.6	2.4	**-1.8**

4xE-Core/1*P-Core = 1.62->1.78x

Engagements:

1) Code Gen (new balanced mode)
2) Improve ISA Execution next revision

E-Cores Shared L2 Savings

P-Core Metric	Stat
Backend_Bound(%)	43.87
..LOAD/STORE_Bound(%)	31.30
....L3_Bound(%)	10.19
......Contested_Accesses(%)	1.98
......Data_Sharing(%)	11.21
......L3_Hit_Latency(%)	79.26

ICC Function	P-Core Clock%	E-Core Clock%	P-E Clock%	P CPI	E CPI	P/E CPI	Notes
Function	**5.82**	**4.33**	**1.49**	**4.05**	**2.93**	**1.38**	Contended accesses

Disasm
vmovdqu ymm0, ymmword ptr [r12+r14*2]
vmovdqu ymmword ptr [r15+r14*2], ymm0
add r14, 10
cmp r14, r8
jb 14029b862

HitCount	Clockticks%	MEM_LOAD_L3_HIT_RETIRED.XSNP_HITM%
5259607889	4.34%	64.05%
5259607889	0.33%	0.00%
5259607889	0.02%	0.00%
5259607889	0.01%	0.00%
5259607889	0.10%	0.00%

E-Cores Appears to Have Cheaper Data Sharing Due to Shared L2 with 1.4x Higher IPC

Summary

- Utilizing a mixture of P-Cores and E-Cores has shown to maximize single threaded, limited threading and MT throughput performance
- Scheduling on the right core is key
  - Intel Thread Director uses core's telemetry to pick the right core to schedule
- Top Down converges performance analysis on E-Cores and P-Cores
  - Can find further opportunities by comparing between the cores
- Targeted optimizations for code generation for E-Cores is delivering further MT gains

Backup