Highlighting the Performance Diversity of Analytical Queries using VOILA

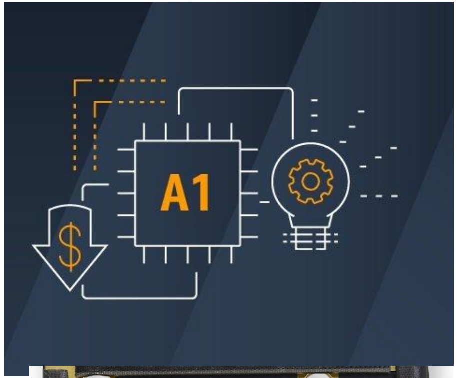
Tim Gubner

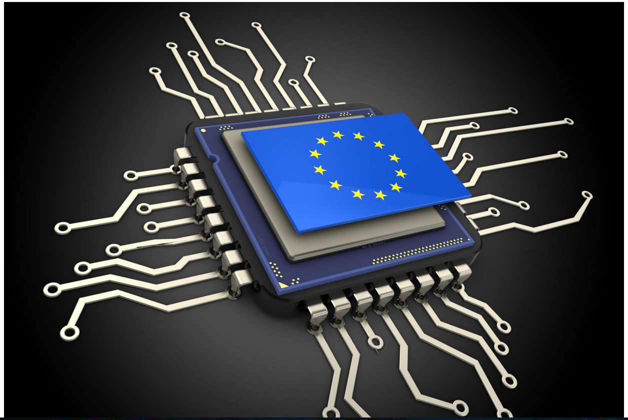
Peter Boncz

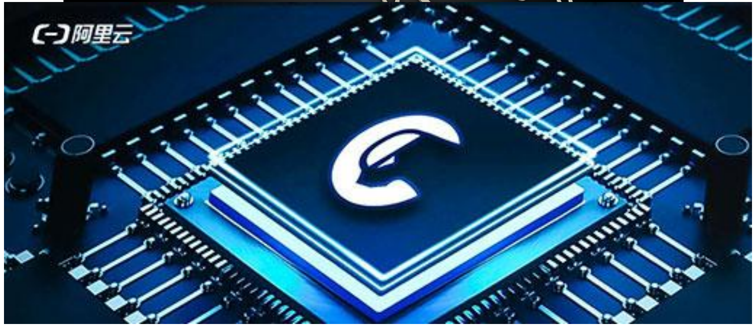
CWI Database Architectures

The X86 space has been quiet lately ...

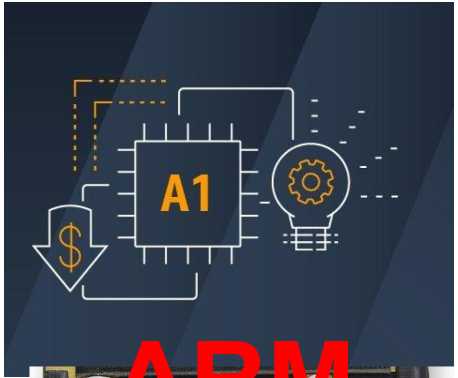
Meanwhile ...

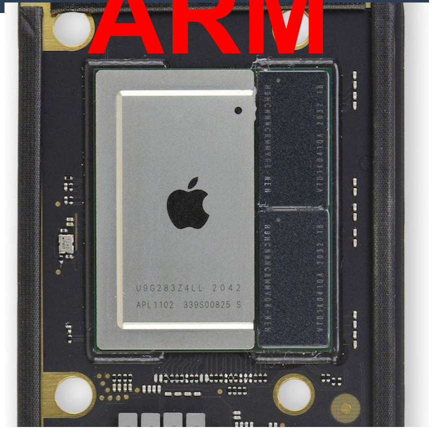




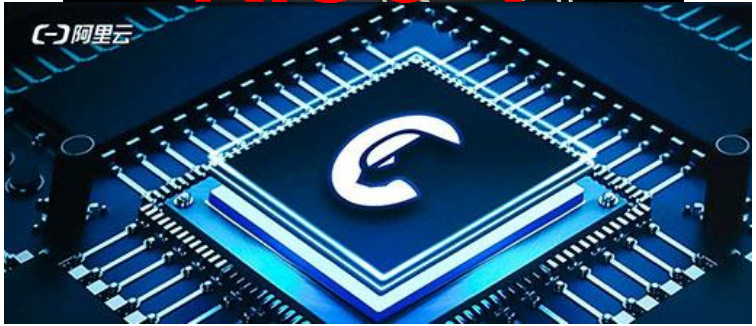


Meanwhile ...





ARM



RISC-V

Why?

Better, more cost-effective

- Money **does** matter
- Optimized chip:
  - License almost-ready chip layout
  - Add specific accelerators
  - Send to some fab
  - Receive *semi-custom tailored chip*

OR

Political/Risk mitigation

- Money **does not** matter
- Independence of US and Intel/AMD
  - Trade wars, sanctions
  - Production/delivery issues

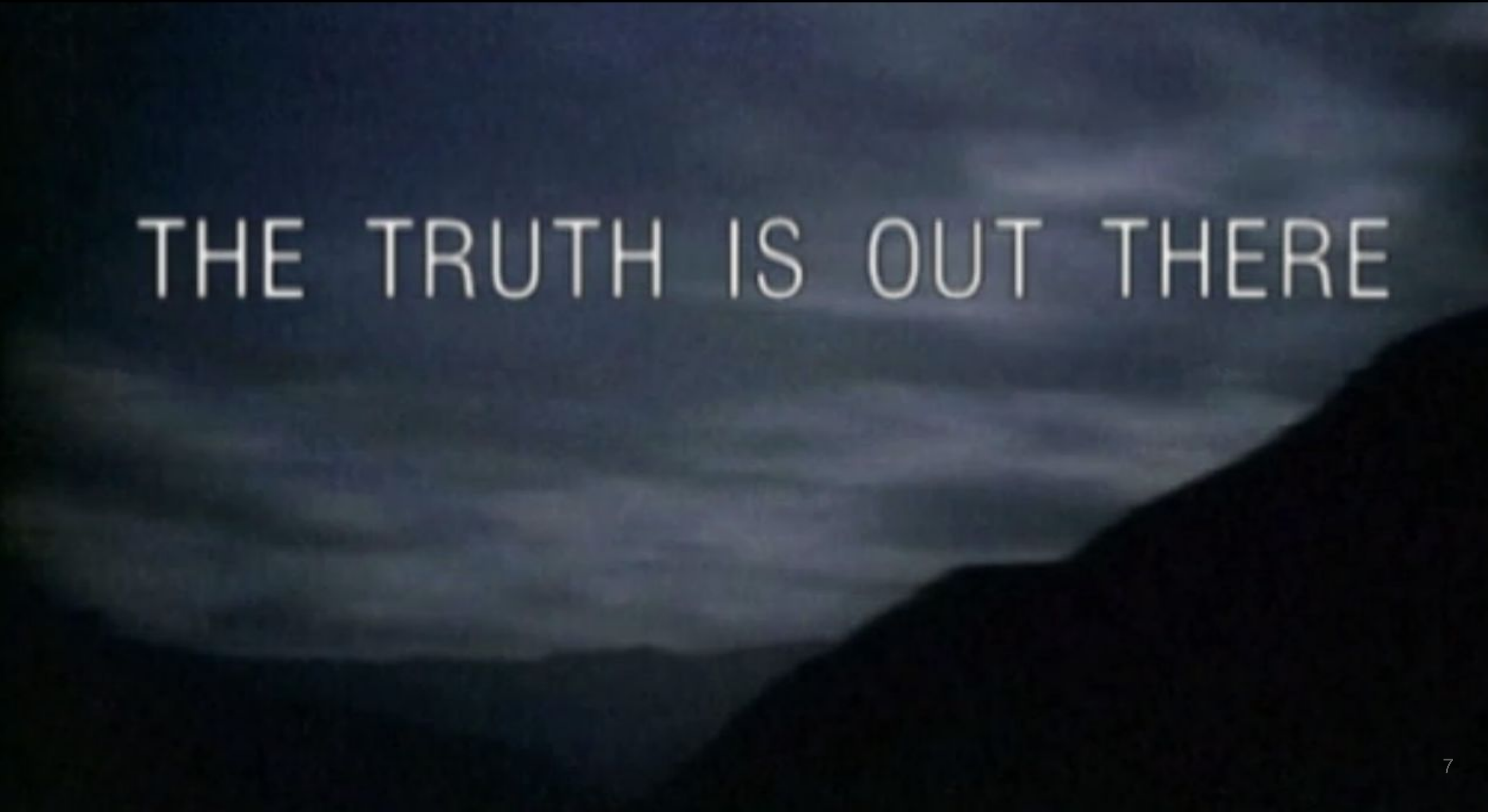
This Work

- Should we optimize for X86?
- Did performance characteristics change?
- Is there performance diversity?

- Is common wisdom (still) valid?
  - ARM is slow*
  - Data-centric outperforms Vectorized on computational workloads^
  - V outperforms DC on data-access-heavy workloads^

* Coffee conversation with senior researcher at CWI

^ Kersten et al. Everything you always wanted to know about compiled and vectorized queries but were afraid to ask. VLDB 2018


THE TRUTH IS OUT THERE

What we did

Ported VOILA-based synthesis framework^ to:

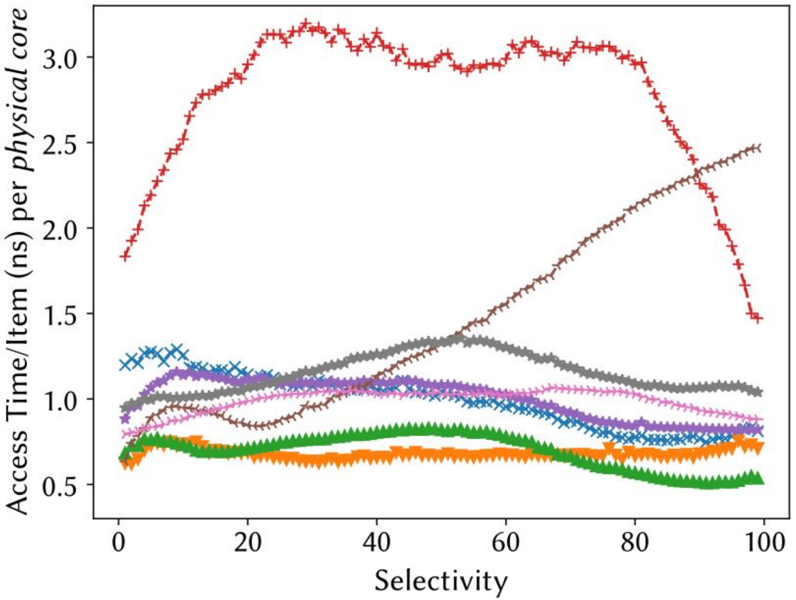
- ARM (M1, Graviton 1, Graviton 2)
- PowerPC (PPC8, PPC9)
- X86 (Skylake-X, 8275C, Epyc)

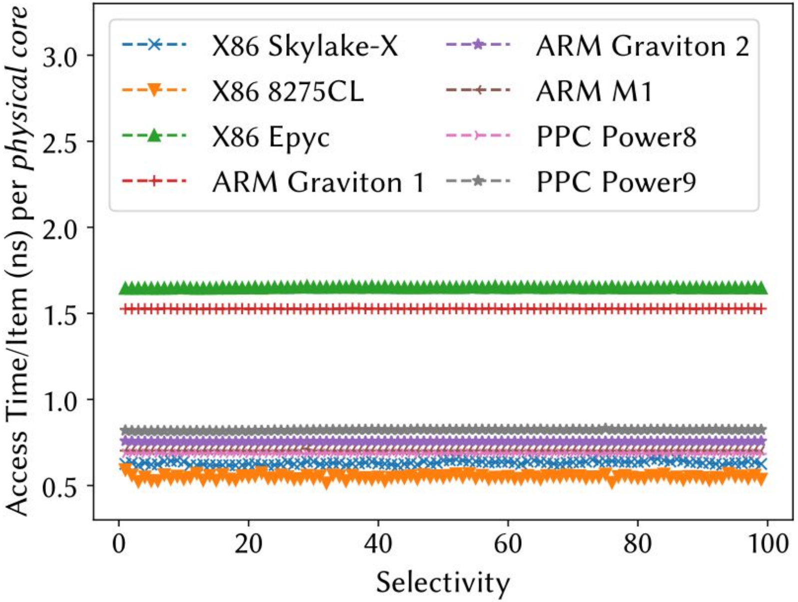
Ran:

- Micro benchmarks (Computation, scalability, control-flow & memory-heavy)
- Macro benchmarks (TPC-H Q1, 3, 6, 9)

^ Gubner, Boncz. Charting the Design Space of Query Execution using VOILA. VLDB 2021. *Thursday Aug 19, R51, around 14:30*

Control-Flow: To Branch, or not to Branch?





Scalability: Does scalable Code scale?

Hardware		Time/Item in ns (slowdown) on given DOP N				
	T	N=1	$T/8$	$T/4$	$T/2$	T
X86 Skylake-X	24	0.08	0.08 (1.0×)	0.09 (1.1×)	0.09 (1.2×)	0.12 (1.5×)
X86 8275CL	96	0.08	0.08 (1.0×)	0.10 (1.4×)	0.17 (2.3×)	*0.18 (2.3×)*
X86 Epyc	96	0.08	0.08 (1.0×)	0.08 (1.0×)	0.12 (1.5×)	*0.17 (2.3×)*
ARM Graviton 1	16	0.34	0.34 (1.0×)	0.36 (1.1×)	0.35 (1.0×)	0.38 (1.1×)
ARM Graviton 2	64	0.20	0.20 (1.0×)	0.20 (1.0×)	0.20 (1.0×)	0.20 (1.0×)
ARM M1	8	0.07	*is* $N = 1$	0.08 (1.1×)	0.08 (1.1×)	0.12 (1.7×)
PPC Power8	128	0.26	0.38 (1.4×)	*0.68 (2.6×)*	*1.27 (4.8×)*	*1.38 (5.2×)*
PPC Power9	128	0.19	0.27 (1.4×)	0.52 (2.7×)	*0.95 (5.0×)*	*1.04 (5.4×)*

Is Vectorization better for Joins?

	Best Flavor	
	Name	best (ms)
X86 Skylake-X	*vec(1024),3,1*	173
X86 8275CL	*vec(512),3,2*	176
X86 Epyc	*scalar,4,16*	134
ARM Graviton 1	scalar,0,1	412
ARM Graviton 2	vec(1024),0,1	101
ARM M1	*vec(1024),4,1*	228
PPC Power8	*vec(512),2,1*	488
PPC Power9	*scalar,3,1*	317

Best Flavor on TPC-H

	Q1	Q3	Q6	Q9
X86 Skylake-X	scalar,0,1	*vec(2048),3,1*	*vec(1024),3,1*	*vec(1024),2,1*
X86 8275CL	*scalar,2,1*	*scalar,3,8*	*vec(512),4,1*	*scalar,2,32*
X86 Epyc	*scalar,2,1*	*vec(256),1,1*	*vec(1024),2,1*	vec(512),0,1
ARM Graviton 1	scalar,0,1	vec(512),0,1	*scalar,4,1*	vec(256),0,1
ARM Graviton 2	*scalar,2,1*	scalar,0,1	*vec(2048),2,1*	vec(512),0,1
ARM M1	scalar,0,1	*vec(2048),2,1*	*scalar,3,1*	*vec(1024),2,1*
PPC Power8	scalar,0,1	*vec(1024),2,1*	vec(256),0,1	*scalar,2,2*
PPC Power9	*scalar,3,1*	vec(512),0,1	*vec(256),2,1*	*scalar,2,8*

Is ARM slow (overall)?

	Q1		Q3	
	x100	hyper	x100	hyper
Runtime (milliseconds)				
X86 Skylake-X	79	54	261	282
X86 8275CL	93	84	480	397
X86 Epyc	81	65	241	238
ARM Graviton 1	188	107	447	447
ARM Graviton 2	**42**	**29**	**162**	**158**
ARM M1	216	86	313	440
PPC Power8	404	384	1094	1132
PPC Power9	239	225	645	631

“Bang for the Buck”

	$\frac{\$}{\text{hour}}$	$\frac{\text{Cents per real core}}{\text{hour}}$	Q9 (ms)	1M × Q9 ($)
X86 Skylake-X (price est.)	1.3392	5.6	228	84
X86 8275CL	0.9122	1.9	232	59
X86 Epyc	0.9122	1.9	193	49
ARM Graviton 1	**0.0788**	**0.5**	720	**16**
ARM Graviton 2	0.7024	1.1	**95**	19

Takeaways

● ARM is slow?	*Not anymore (faster, cheaper)*
● V outperforms DC on data-access?	*Sometimes*
● DC outperforms V on computation?	*Yes*
● Branching outperforms data-dependency?	*Sometimes*
● Scalable programs scale?	*Sometimes*

Performance diversity is real: adapt or perish.

The best days of X86 are over.