Quantifying the Effects of Copious 3D-Stacked Cache on HPC Workloads

Emil Vatai, Jens Domke, Balazs Gerofi, Yuetsu Kodama, Mohamed Wahib, Artur Podobas, Sparsh Mittal, Miquel Pericàs, Lingqi Zhang, Peng Chen, Aleksandr Drozd, Satoshi Matsuoka

RIKEN Center for Computational Science, Intel Corporation, KTH Royal Institute of Technology, Indian Institute Of Technology, Roorkee, Chalmers University of Technology, Tokyo Institute of Technology, National Institute of Advanced Industrial Science and Technology

Introduction

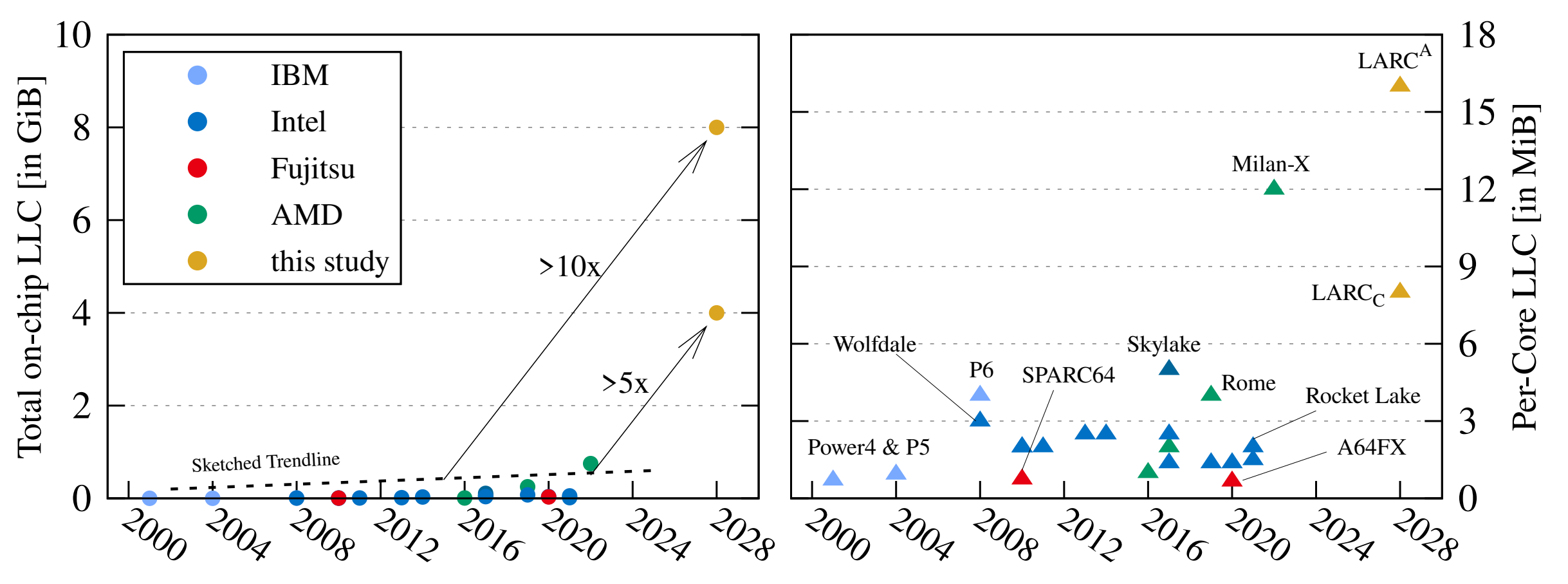


Figure 1: A sample of representative server-grade CPUs of each generational micro-architecture in comparison to our study of LARC; Left: total on-chip last-level cache (in GiB); Right: per-core last-level cache (in MiB) for the same CPUs; The two LARC variants will be discussed in detail in "Gem5 Config".

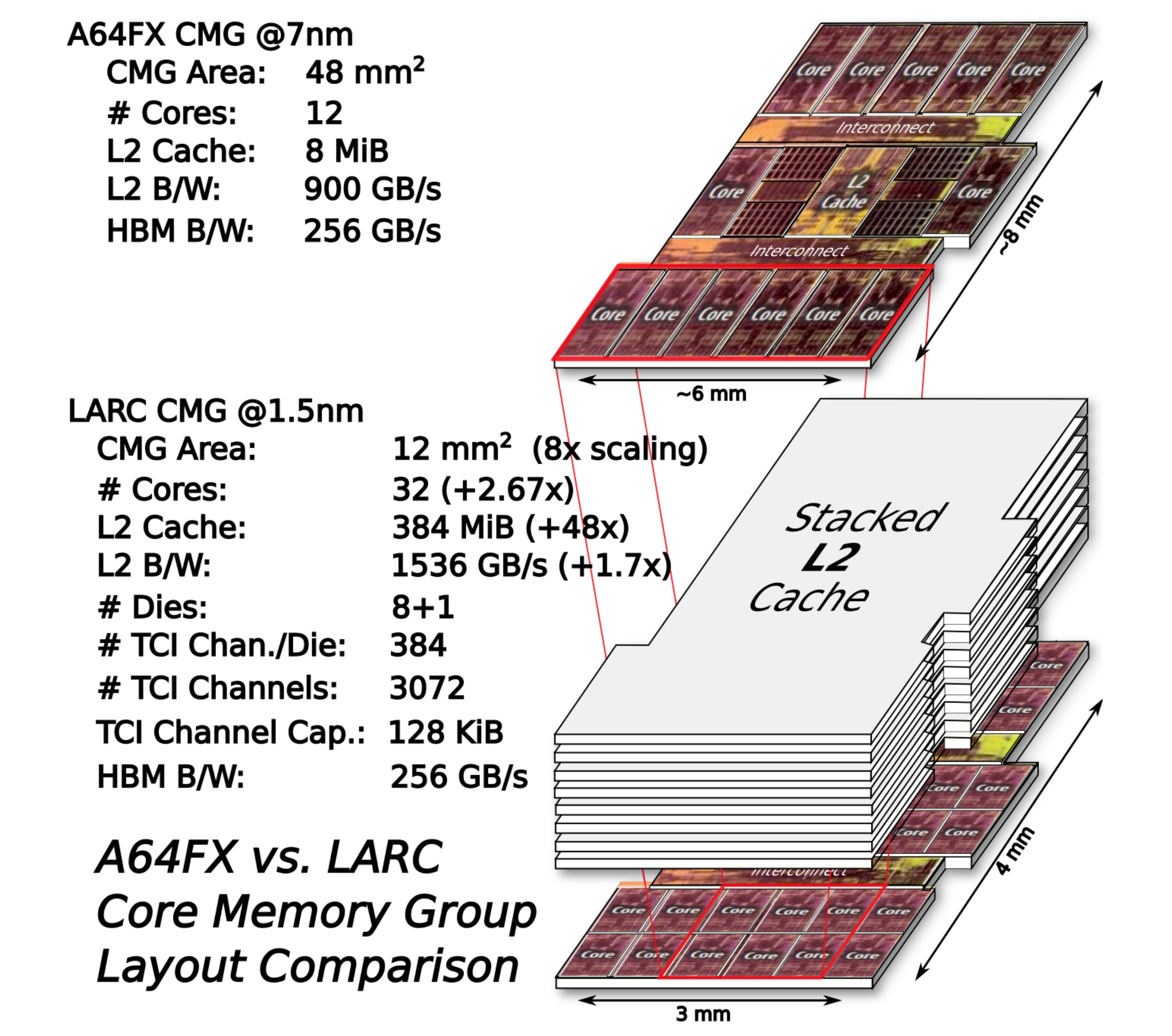
Design of a **novel exploration framework** (MCA) that allows us to simulate HPC applications running on a hypothetical processor having **infinitely large L1D cache** and is **orders of magnitude faster** than cycle-accurate simulators, and is used to estimate an upper-bound for cache-based improvements.

We **model a hypothetical LARge Cache processor (*LARC*)**, that builds on the design of A64FX, with an LLC (Last Level Caches) designed with eight stacked SRAM dies under 1.5 nm manufacturing assumption.

We simulate the performance of the LARC processor on a plethora of **simulations of HPC proxy-applications** using the cycle-accurate **Gem5 simulator**.

The hypothetical LARC

The hypothetical LARC processor is modelled by after the A64fx CPU, by fitting **16 CMGs, each with 32 cores** on the same die size, combining it with 384 MiB 3D-stacked SRAM (per CMG).



A64FX CMG @7nm

CMG Area:	48 mm^2
# Cores:	12
L2 Cache:	8 MiB
L2 B/W:	900 GB/s
HBM B/W:	256 GB/s

LARC CMG @1.5nm

CMG Area:	12 mm^2 (8x scaling)
# Cores:	32 (+2.67x)
L2 Cache:	384 MiB (+48x)
L2 B/W:	1536 GB/s (+1.7x)
# Dies:	8+1
# TCI Chan./Die:	384
# TCI Channels:	3072
TCI Channel Cap.:	128 KiB
HBM B/W:	256 GB/s

Figure 2: Difference between A64FX's Core Memory Group (CMG) and a LARC CMG in various performance-governing parameters; Most notable (for our study) is the 48x increase in per-CMG L2 cache capacity; **Note:** despite appearing similar in the figure, the LARC CMG is, in fact, four times smaller.

Paper

At the Locus of Performance: A Case Study in Enhancing CPUs with Copious 3D-Stacked Cache

`https://arxiv.org/abs/2204.02235`

Gem5 config

RIKEN's fork of the Gem5 cycle-accurate simulator **supports SVE and HBM.** Considering the restrictions of Gem5 we use run simulations with a **conservative** ($LARC_C$) and an aggresive **$LARC^A$** configuration; $LARC_S$ and $LARC^{32}$ are Gem5 configurations for validation.

	$A64FX_S$	**$A64FX^{32}$**	**$LARC_C$**	**$LARC^A$**
Cores	12	32	32	32
CMGs	4	4	16	16
Core config.	Arm v8.2 + SVE, 512 bit SIMD, 2.2 GHz, OoO 128 ROB entries, dispatch width 4			
Branch pred.	Bi-mode: 16 K global predictor, 16 K choice predictor			
Per-core L1D	64 KiB 4-way set-assoc, 3 cycles adjacent line prefetcher			
	L2 CACHE PER CMG:			
L2 size	8 MiB		256 MiB	512 MiB
BW	~ 800 GB/s		~ 800 GB/s	~ 1600 GB/s
	L2 CACHE AGGREGATED:			
L2 size	32 MiB		4096 MiB	8192 MiB
BW	~ 3.2 TB/s		~ 12.8 TB/s	~ 25.6 TB/s
L2 config.	16-way set-associative, 37 cycles, inclusive, 256 B block			
Main Memory	32 GiB HBM2, 4 channels, 256 GB/s			

Table 1: Chip area and simulator configurations for gem5

SDE+MCA framework

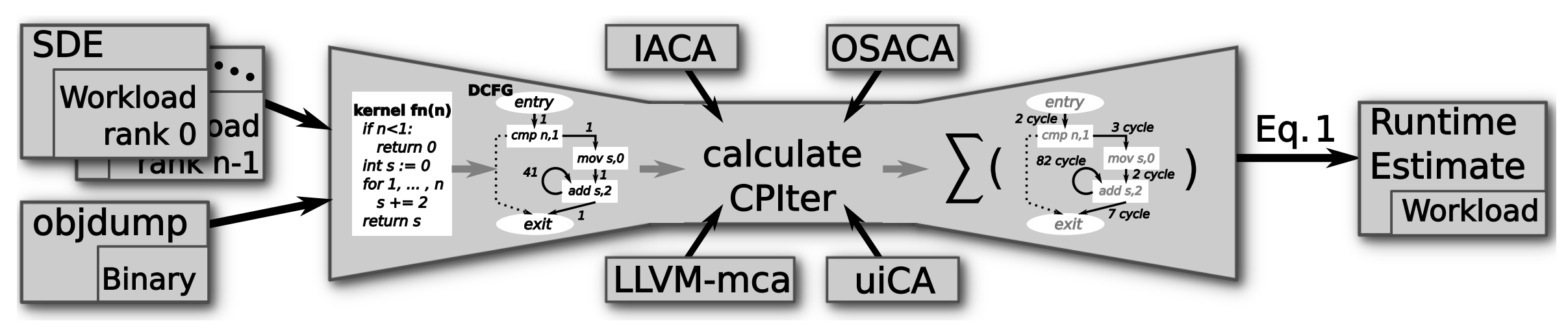


Figure 3: Combining the output of SDE and MCA to estimate the runtime.

Intel Software Development Emulator (SDE) can record:

- the basic blocks and their caller/callee dependencies;
- the number of invocations per control flow graph (CFG) edge, i.e., how often the program counter (PC) jumped from one specific basic block to another;
- the Assembly code of the basic blocks;

From the Assembly code, Machine Code Analyzer (MCA) can approximate the cycles-per-iteration (CPIter) of each basic block. We estimate the runtime of the workload by combining this information (considering multiple OpenMP and MPI ranks) as follows (Figure 3):

$$t_{\text{app}} := \frac{\max\limits_{r \in \text{ranks}} \left(\max\limits_{t \in \text{threads}_r} \left(\sum\limits_{\text{edges } e \in \text{CFG}_{t,r}} \text{CPIter}_e \cdot \#\text{calls}_e \right) \right)}{\text{processor frequency in Hz}}$$

Results

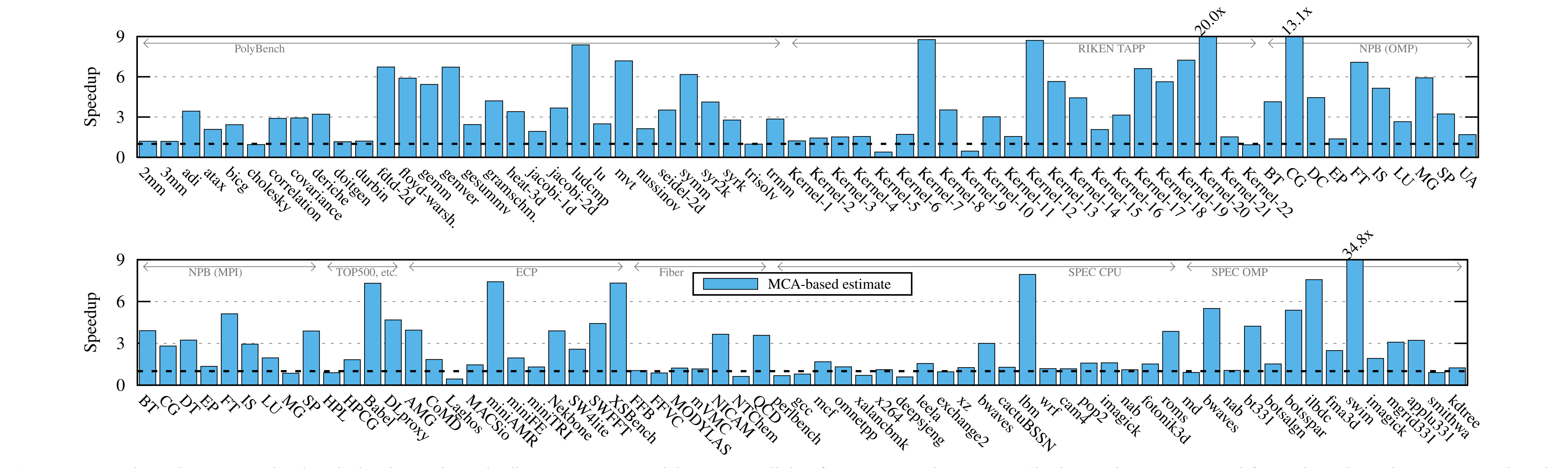


Figure 4: Projected speedup against a baseline dual-socket Intel Broadwell E5-2650v4 system while assuming all data fits into L1D with "optimistic" load-to-use latency; Top row, left to right: PolyBench, RIKEN TAPP kernels, NPB (OMP); Bottom row, left to right: NPB (MPI), TOP500 etc., ECP proxies, RIKEN Fiber apps, SPEC CPU[int/single] and CPU[float/OMP], SPEC OMP

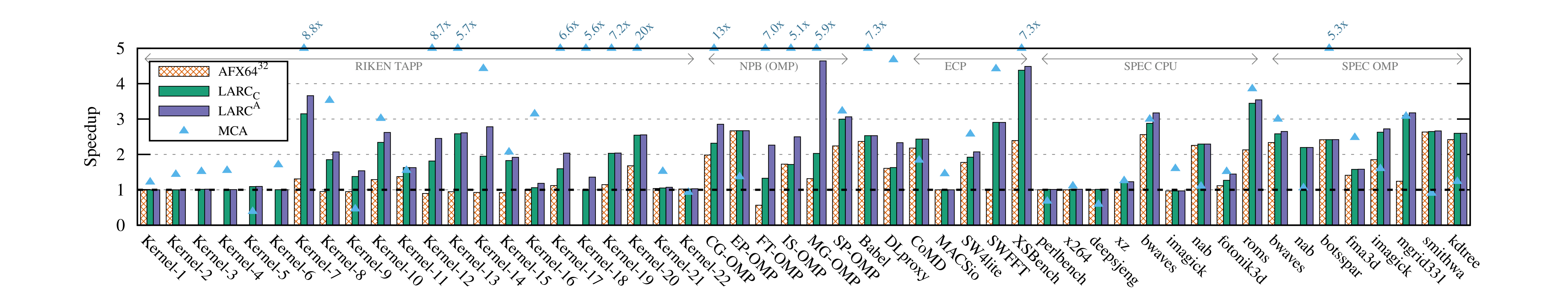


Figure 5: gem5-based, simulated speedups of $A64FX^{32}$, $LARC_C$ and $LARC^A$ in comparison to baseline $A64FX_S$; Left to right: RIKEN TAPP kernels, NPB (OMP), TOP500 etc., ECP proxies, SPEC CPU[int/single] and CPU[float/OMP], SPEC OMP; Added MCA-based estimations from Fig. 4 for reference; Kernel 3–6 and 18 limited to 12 threads, hence we omit $A64FX^{32}$; Missing benchmarks (cf. Fig. 4) primarily due to gem5 issues or exceeding simulation time limit. PolyBench results (single core) are also omitted due to limited speedup across all of them and no noteworthy outliers.

Validation

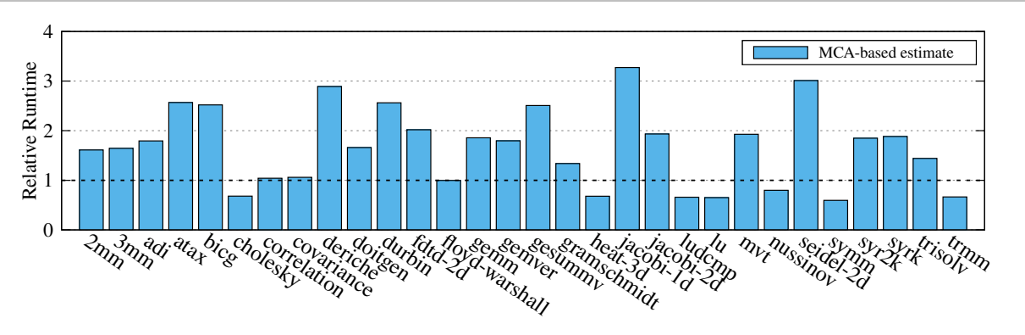


Figure 6: Validation of MCA-based performance estimation against PolyBench/C `MINI` with inputs fitting into L1D; Relative runtime shown (vs. Intel E5-2650v4 measurements); Values ≥ 1 show prediction of faster execution

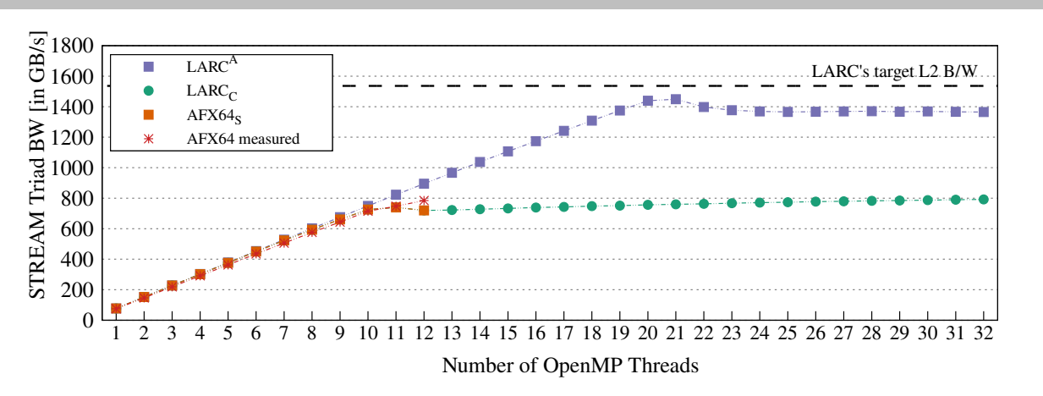


Figure 7: Validation of the simulated STREAM Triad bandwidth for fixed 128 KiB vectors per core; $A64FX_S$ scaled to 12 cores; Real A64FX measurements on 1 CMG for reference; Dashed lines highlight trend (not measured)

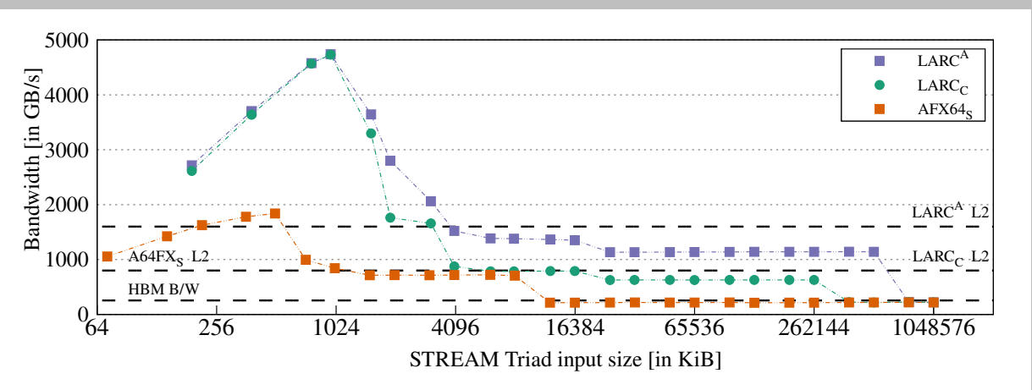


Figure 8: Validation of the simulated STREAM bandwidth for both LARC configurations with 32 cores (vs. $A64FX_S$ with 12 cores); STREAM Triad input range from few KiB to 1 GiB; Dashed lines show trend (not measured)

Conclusion

- Over half (**31 out of 52**) of the simulated applications experience a $\geq$ **2x speedup** on LARC's Core Memory Group (CMG) that occupies only one fourth the area of the baseline A64FX CMG.
- For applications that are responsive to larger cache capacity, this would translate to an **average improvement of 9.56x** (geometric mean) when we assume ideal scaling and compare at the full chip level.

HPC (Proxy-)Apps and Benchmarks

Polyhedral Benchmark Suite 30 single-threaded, scientific kernels (the largest configuration is used).
TOP500 HPL (dense), HPCG (sparse) solvers.
BabelStream memory subsystem evaluation.
DLproxy micro-benchmark representing 2D deep CNNs.

NASA Advanced Supercomputing Parallel Benchmarks 9 kernels and proxy-apps common in CFD.
RIKEN's Fiber Mini-Apps and TAPP Kernels representing the scientific priority areas of Japan and the scaled-down versions tailored for fast simulations with gem5.

Exascale Computing Project Proxy-Applications a co-design benchmarking suite curated by US-based supercomputing centers.
SPEC CPU & SPEC OMP Benchmarks two HPC-focused benchmark suits: SPEC CPU[speed] and SPEC OMP.