Is RISC-V ready for HPC prime-time:

EVALUATING THE 64-CORE SOPHON SG2042 RISC-V CPU

Nick Brown, Maurice Jamieson, Joseph Lee (EPCC)

Paul Wang (PerfXLab)

Outline

- SG2042 vs other RISC-V commodity hardware
- Lessons learnt from optimizing for SG2042
- SG2042 vs x86 CPUs

RISC-V for HPC?

- Currently:
  - HPC as strategic priority area for growth
  - Lots of activity with RISC-V HPC *software*
  - But HPC community can only access embedded Single Board Computers (apart from a few prototypes) or soft-cores
  - → 64-core Sophon SG2042 CPU

epcc

Sophon SG2042

- First publicly available, mass-produced **64-core RISC-V CPU**
- Aimed at high-performance workloads
- T-Head XuanTie C920 core for high performance
- ISA: RV64GCV
- RVV v0.7.1 – 128-bit vector

Sophon SG2042 Tech Specs

Clock speed	2GHz
Pipeline	12-stage out-of-order
Execution units	3 decode 4 rename/dispatch 9 issue/execute 2 load/store
Cache	64KB L1 instruction (I) and data (D) cache 1MB L2 shared between 4 cores 64MB L3 shared by all cores
Connectivity	32 lanes PCI-E Gen4
Memory	4 DDR4-3200 memory controller (32GB in Milk-V Pioneer Box – available up to 128GB)

SG2042 vs other RISC-V processors

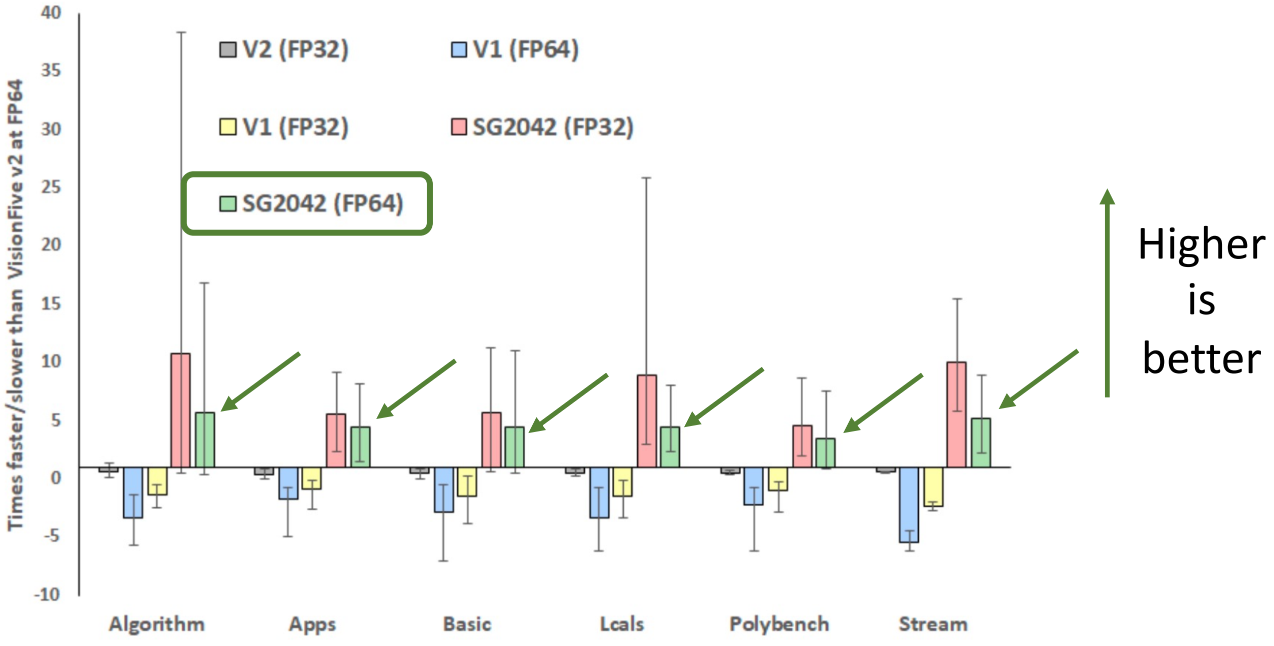
	StarFive VisionFive V1	StarFive VisionFive V2	Sophon SG2042
SoC	JH7100	JH7110	SG2042
Cores	SiFive U74 (x2)	SiFive U74 (x4)	XuanTie C920
Clock Speed	1.2 GHz	1.5 GHz	2.0 GHz
Cache	32KB (D) + 32KB (I) L1 2MB L2 shared between all cores	32KB D + 32KB I L1 2MB L2 shared between all cores	64KB (I) and (D) L1 1MB L2 shared between 4 cores 64MB L3 shared by all cores
Instruction Set	RV64GC	RV64GC	RV64GC+V0p7

RAJAPerf

- Benchmark: RAJA Performance Suite (https://github.com/LLNL/RAJAPerf)
  - ALGORITHM
  - APPS
  - BASIC
  - LCALS (Livermore Compiler Analysis Loop Suite)
  - POLYBENCH
  - STREAM

SG2042 vs other RISC-V processors

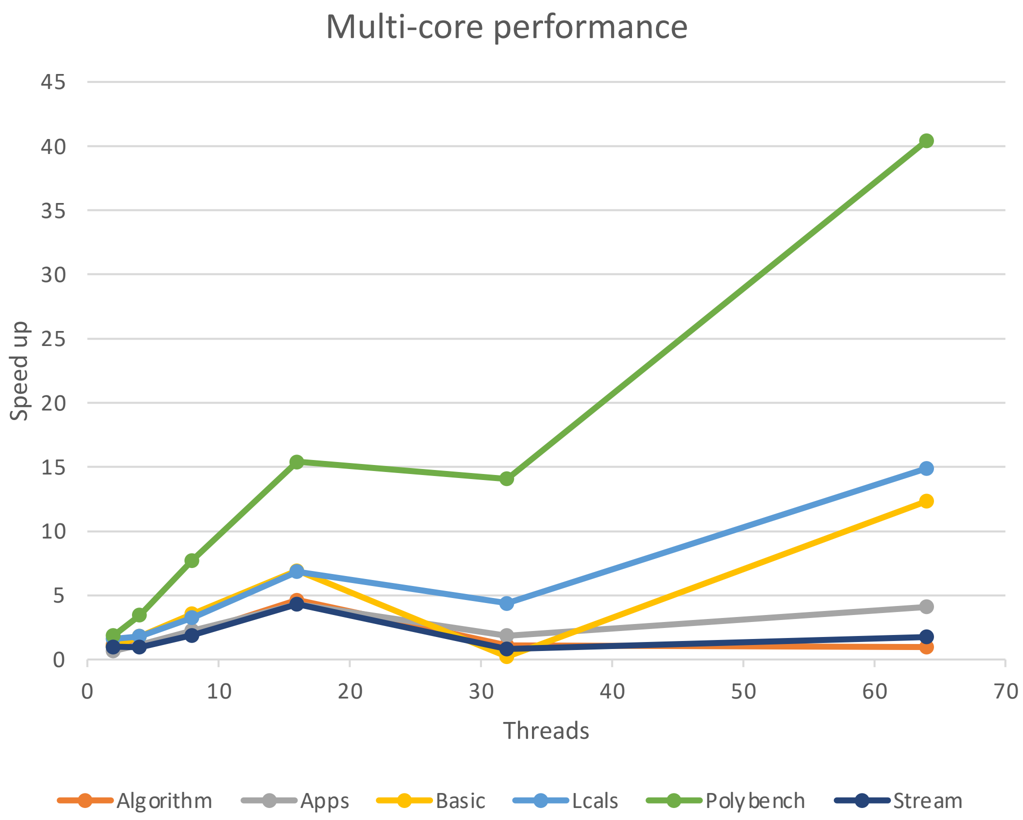
- Bars report an average across the specific RAJAPerf class
- Baseline = VisionFive V2 with FP64
- 0 = same performance
- +1 = 1x faster
- -1 = 1x slower
- Single Core Result:
  - SG2042 outperforms VisionFive V2
  - FP64: 4.3x to 6.5x faster than U74 in V2



Single core comparison: SG2042 vs VisionFive V1 and V2

Squeezing Performance from SG2042

- Multi-core performance
- OpenMP: OMP_PROC_BIND
- Most kernels only beneficial at small thread count
- Default: Generally mediocre parallel efficiency

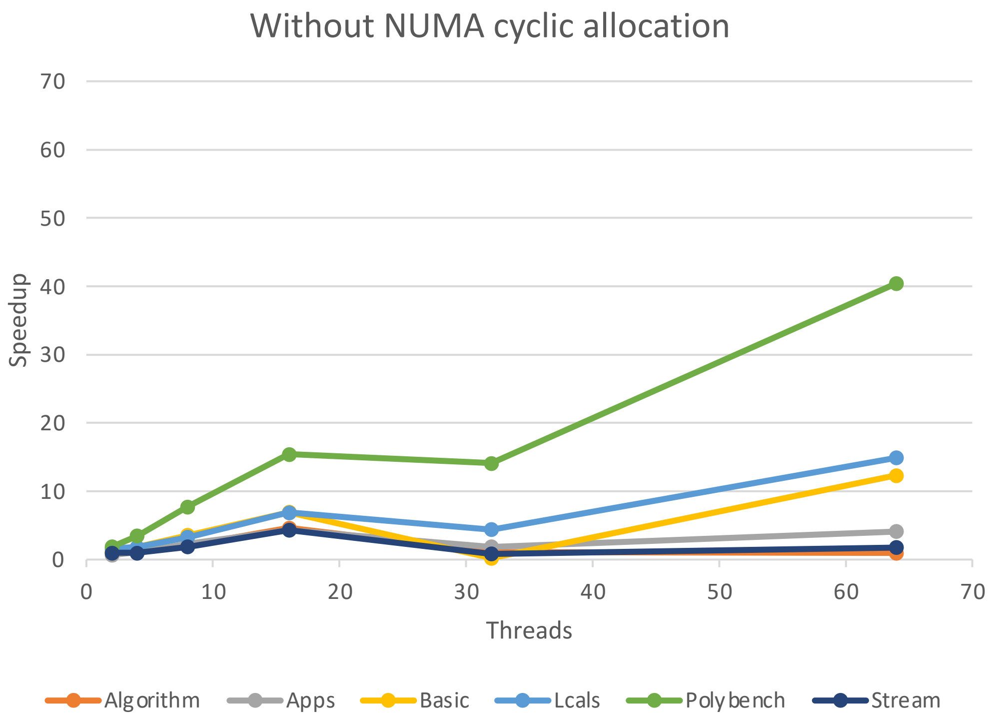


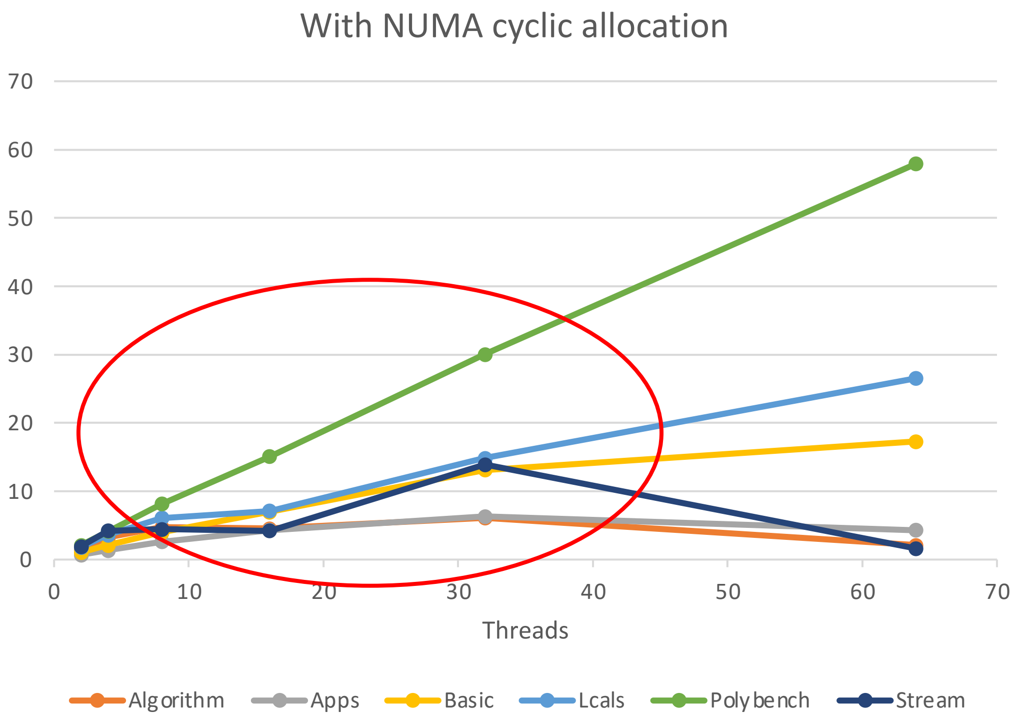
FP32

Squeezing Performance

0-3	4-7	8-11	12-15
16-19	20-23	24-27	28-31
32-35	36-39	40-43	44-47
48-51	52-55	56-59	60-63

- 4 NUMA regions, each with memory controller
- Core ids not contiguous in NUMA region
- Allocate cyclically: 0 → 8 → 32 → 40 → 1 → 9 → 33 → 41
- Significant improvement



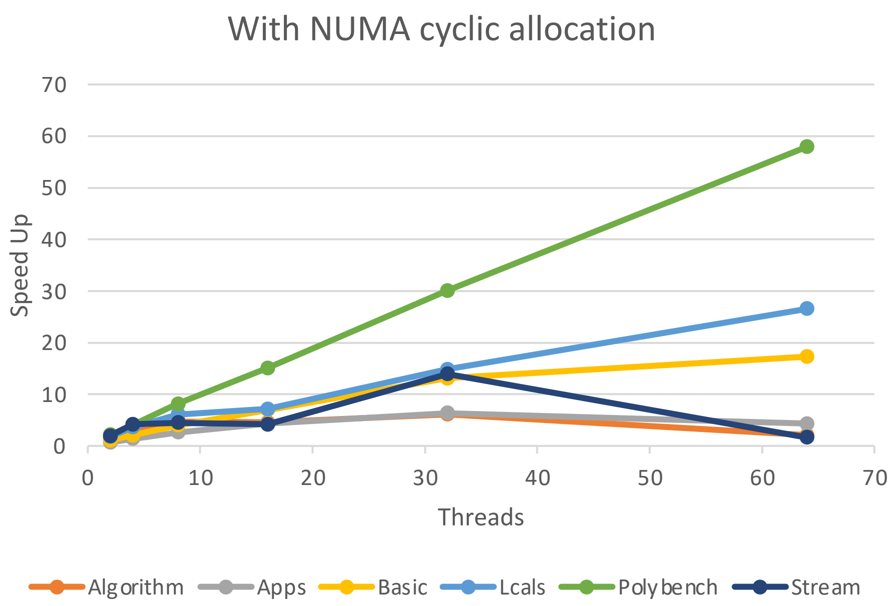


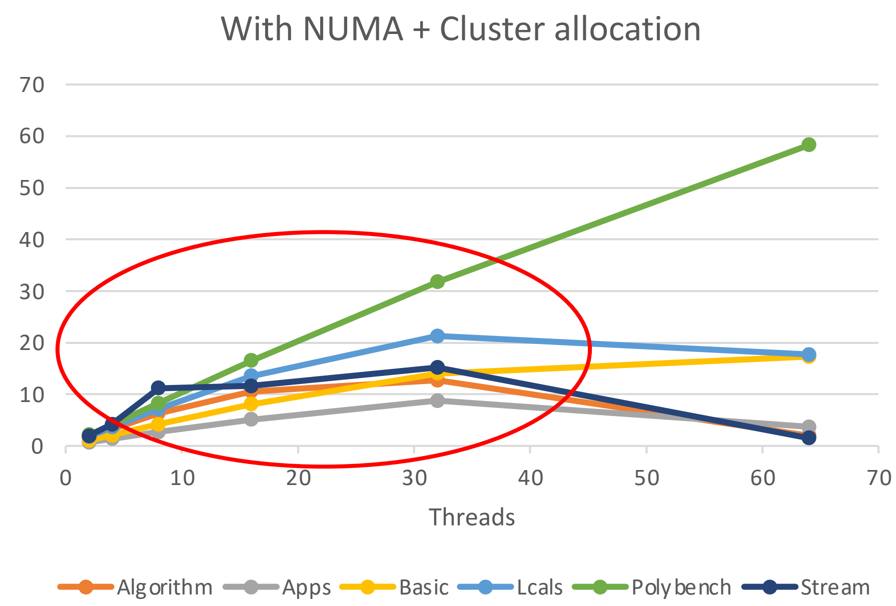
epcc

Squeezing Performance

0-3	4-7	8-11	12-15
16-19	20-23	24-27	28-31
32-35	36-39	40-43	44-47
48-51	52-55	56-59	60-63

- 4 cores share L2 memory
- NUMA region + L2 cluster
- Allocate with cluster: 0 → 8 → 32 → 40 → 16 → 24 → 48 → 56
- Further improvement in medium thread count

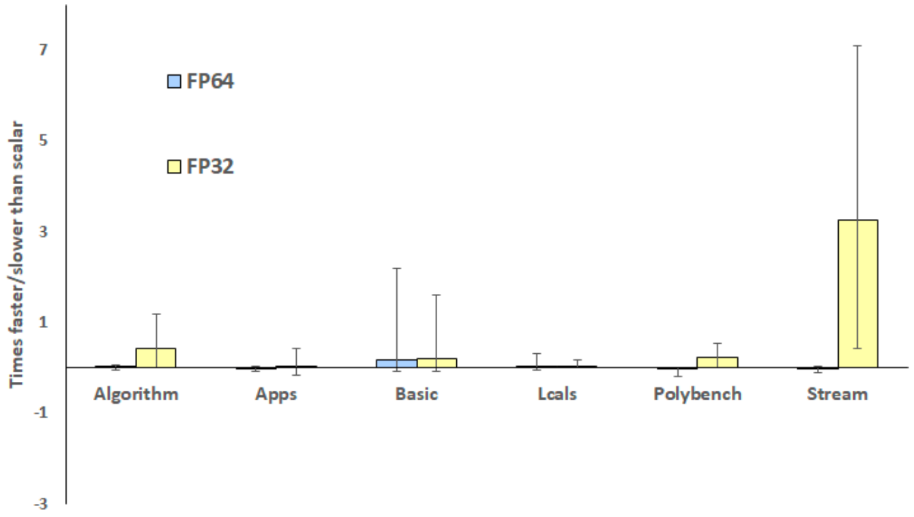




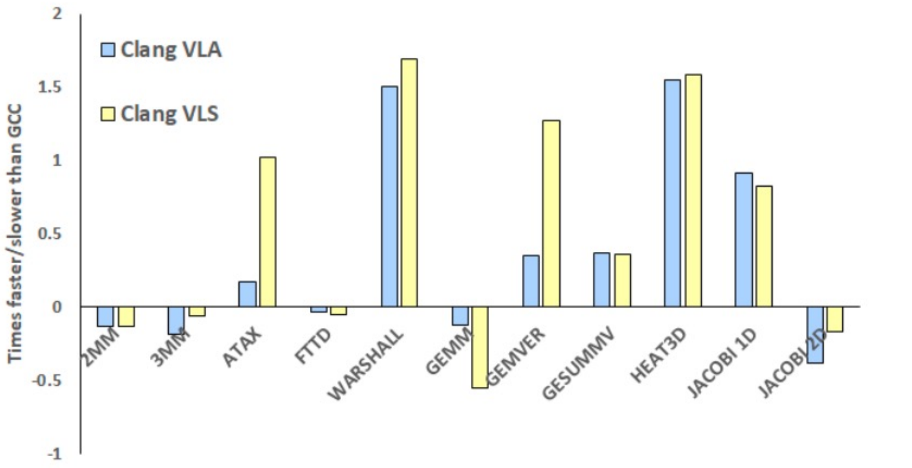
epcc

Squeezing performance

- Vectorization:
  - More significant benefit for FP32 than FP64
  - 128-bit vector length
  - Vary significantly for different kernels
- Compiler:
  - Clang vs GCC makes a difference
  - Clang also supports Vector Length Agnostic / Specific (VLA/VLS)
  - Clang only supports RVV 1.0, require tool to backport to RVV 0.7 (arXiv: 2304.10324)



Single core speed-up with vectorisation



Compiler vectorization comparison: Clang VLA/VLS vs GCC

Comparison against x86

CPU	Part	Clock	Cores	Vector
AMD Rome	EPYC 7742	2.25GHz	64	AVX2 (256-bit)
Intel Broadwell	Xeon E5-2695	2.1GHz	18	AVX2 (256-bit)
Intel Icelake	Xeon 6330	2.0GHz	28	AVX512 (512-bit)
Intel Sandybridge	Xeon E5-2609	2.40GHz	4	AVX (128-bit)

- Use most performant number of threads: for x86 = number of cores
- Bind cores & disable hyperthreading
- GCC 8.3 for Intel systems, GCC 11.2 for AMD Rome
- Compiled with optimisation –O3

Comparison against x86

- Single core:
  - On average, all x86 tested outperform SG2042 (C920 core)
  - Except Sandybridge: slower for *stream* and *algorithm*
  - For some kernels, C920 faster

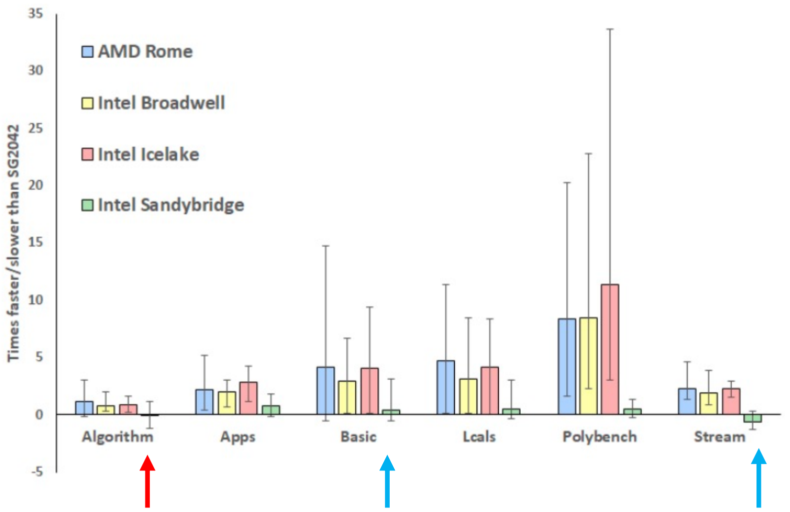


Figure 4: FP64 single core comparison against x86, reporting number of times faster or slower than the baseline SG2042

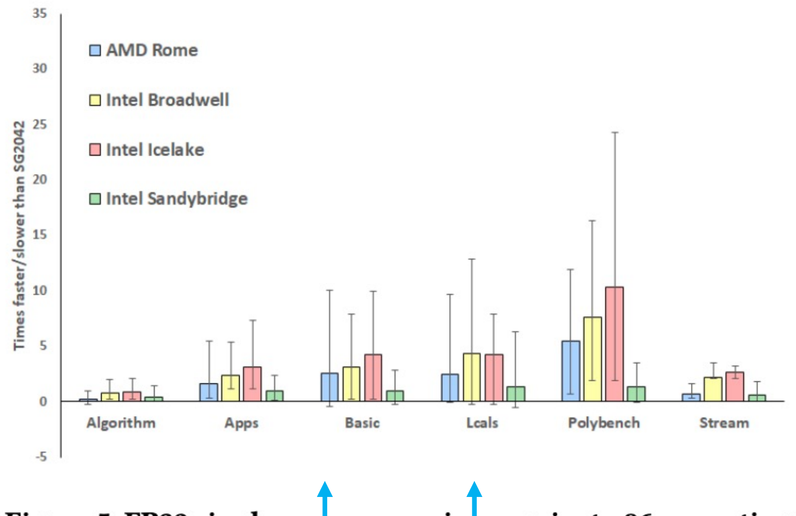


Figure 5: FP32 single core comparison against x86, reporting number of times faster or slower than the baseline SG2042

Comparison against x86

- Multi-thread:
  - SG2042 takes advantage of greater core-count, outperforms Sandybridge (4 core)
  - AMD Rome, Intel Broadwell, Icelake outperforms SG2042 for most kernels
  - In many cases, by a significant amount

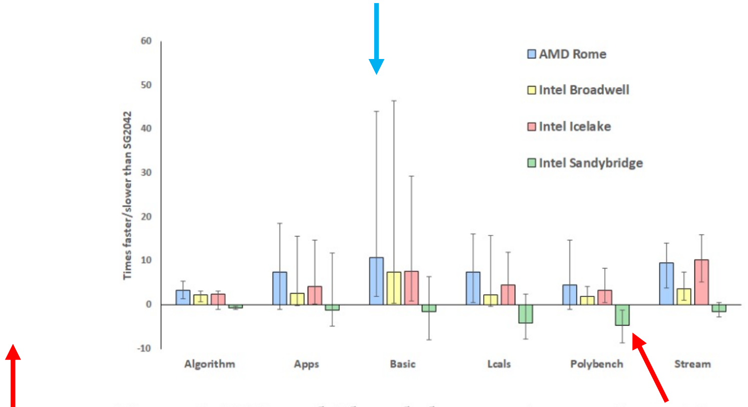


Figure 6: FP64 multithreaded comparison against x86, reporting number of times faster or slower than the baseline SG2042

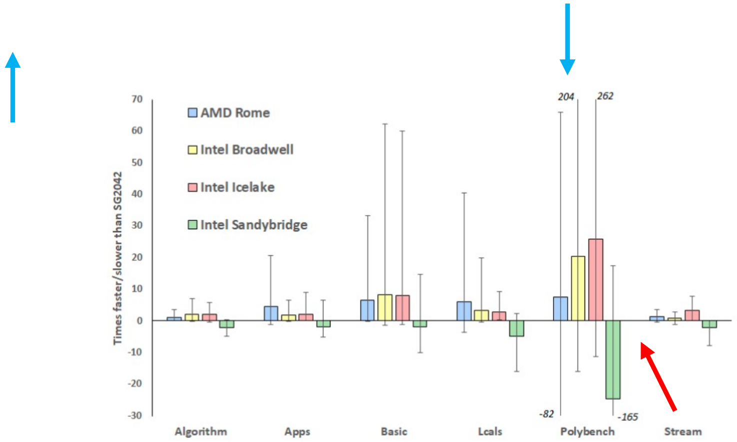


Figure 7: FP32 multithreaded comparison against x86, reporting number of times faster or slower than the baseline SG2042

Conclusion

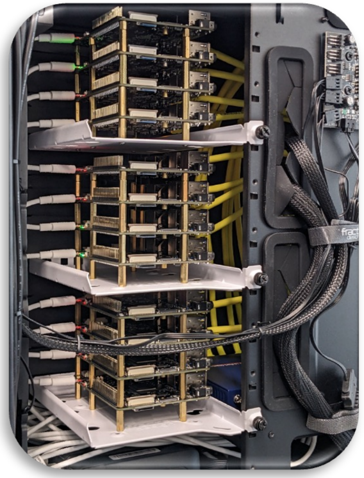
- SG2042 presents first publicly available 64-core RISC-V processor
- It has better single core performance than other RISC-V SBC
- Explored optimisation via thread placement
- Compared against x86 CPUs
- Further questions:
    - MPI?

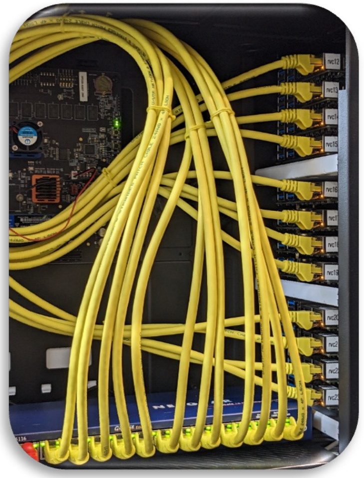
- Exciting future for HPC!

EPCC RISC-V Testbed

- Aim: Provide HPC code developers and data-scientists access to the latest RISC-V CPUs
- We have many boards:

Development Board	Processor (SOC)	# Core	Qty
NezhaSTU	C906 (D1)	1	4
MangoPi MQ-Pro	C906 (D1)	1	2
HiFive Unmatched	U74 (FU740)	4	1
StarFive VisionFive V1	U74 (JH7100)	2	3
StarFive VisionFive V2	U74 (JH7110)	4	13
Milk-V Pioneer	C920 (SG2042)	64	2





Soft Cores (on FPGAs)

- NEORV32
- Andes University Program

Blog posts with experience building RISC-V

Thank you!

- EPCC RISC-V Testbed: http://riscv.epcc.ed.ac.uk/

Comparison against x86 (Backup)

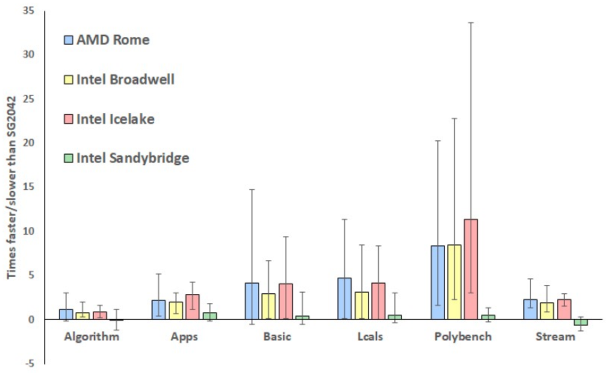


Figure 4: FP64 single core comparison against x86, reporting number of times faster or slower than the baseline SG2042

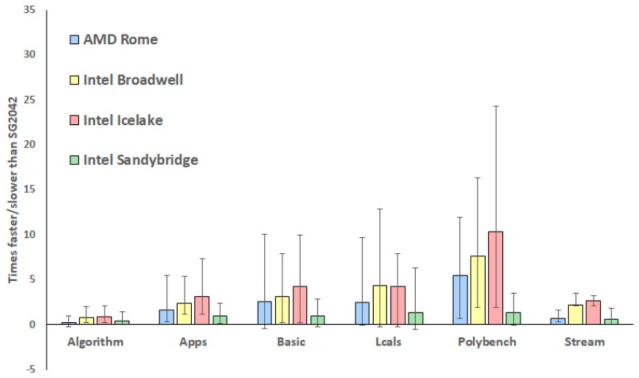


Figure 5: FP32 single core comparison against x86, reporting number of times faster or slower than the baseline SG2042

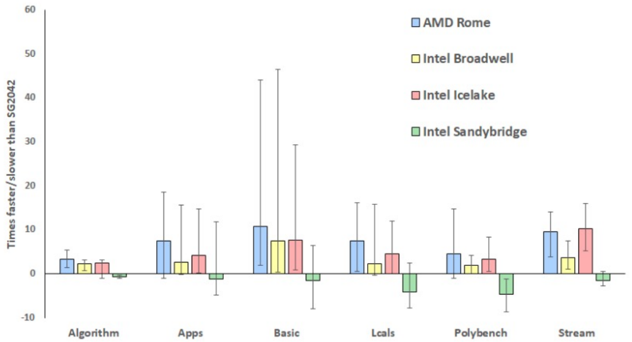


Figure 6: FP64 multithreaded comparison against x86, reporting number of times faster or slower than the baseline SG2042

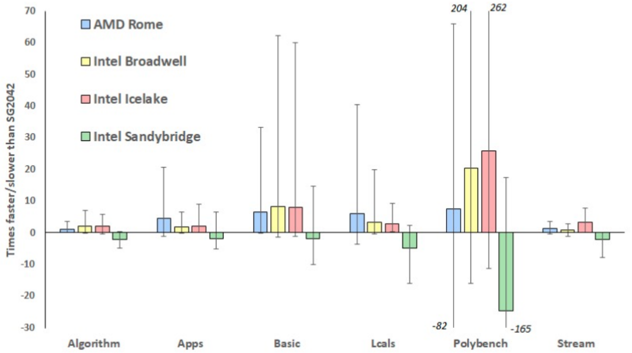


Figure 7: FP32 multithreaded comparison against x86, reporting number of times faster or slower than the baseline SG2042