Programming the Cell Processor: Achieving High Performance and Efficiency

Presented by

Jeremy S. Meredith
Sadaf R. Alam
Jeffrey S. Vetter

Future Technologies Group
Computer Science and Mathematics Division

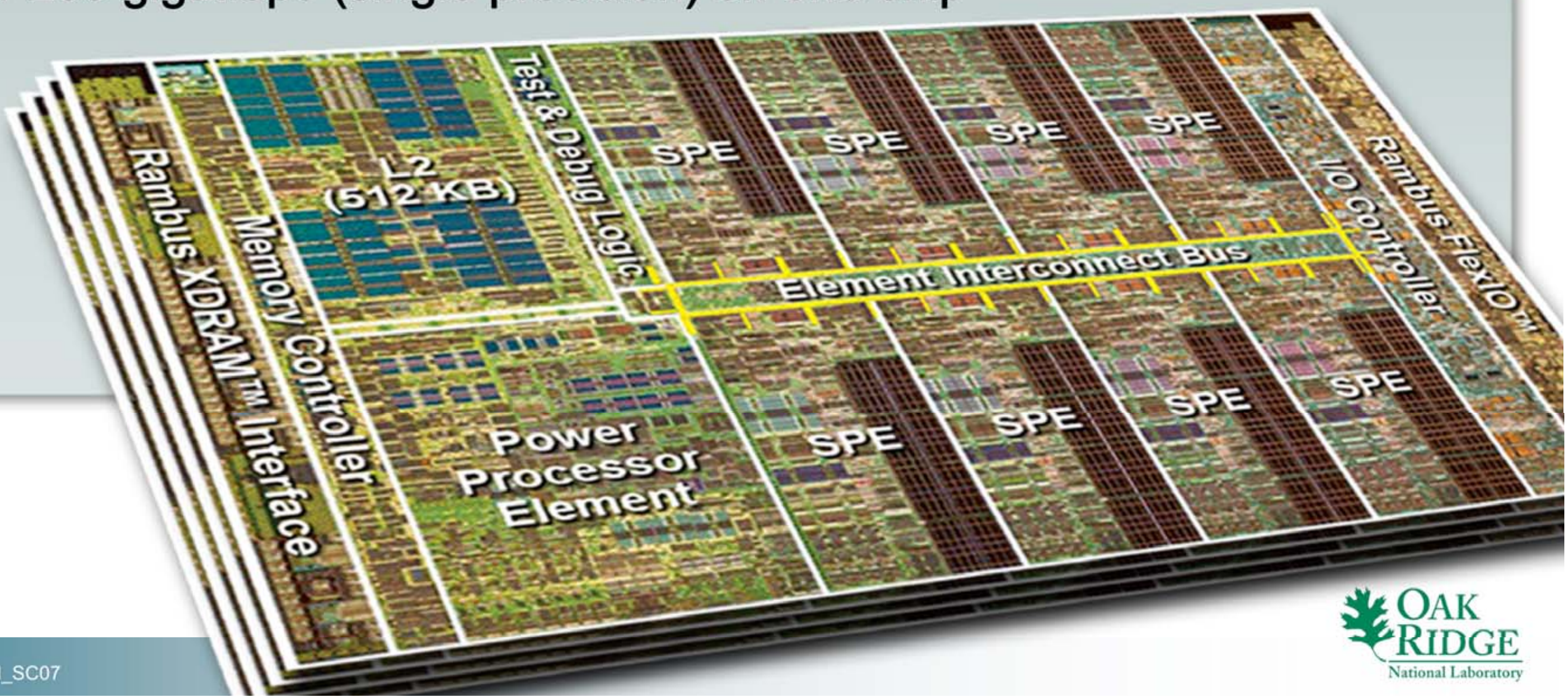
Research supported by the Department of Energy's Office of Science
Office of Advanced Scientific Computing Research

SC07

OAK RIDGE National Laboratory

Cell broadband engine processor: An overview

- One POWER architecture processing element (PPE)
- Eight synergistic processing elements (SPEs)
- All connected through a high-bandwidth element interconnect bus (EIB)
- Over 200 gigaflops (single precision) on one chip



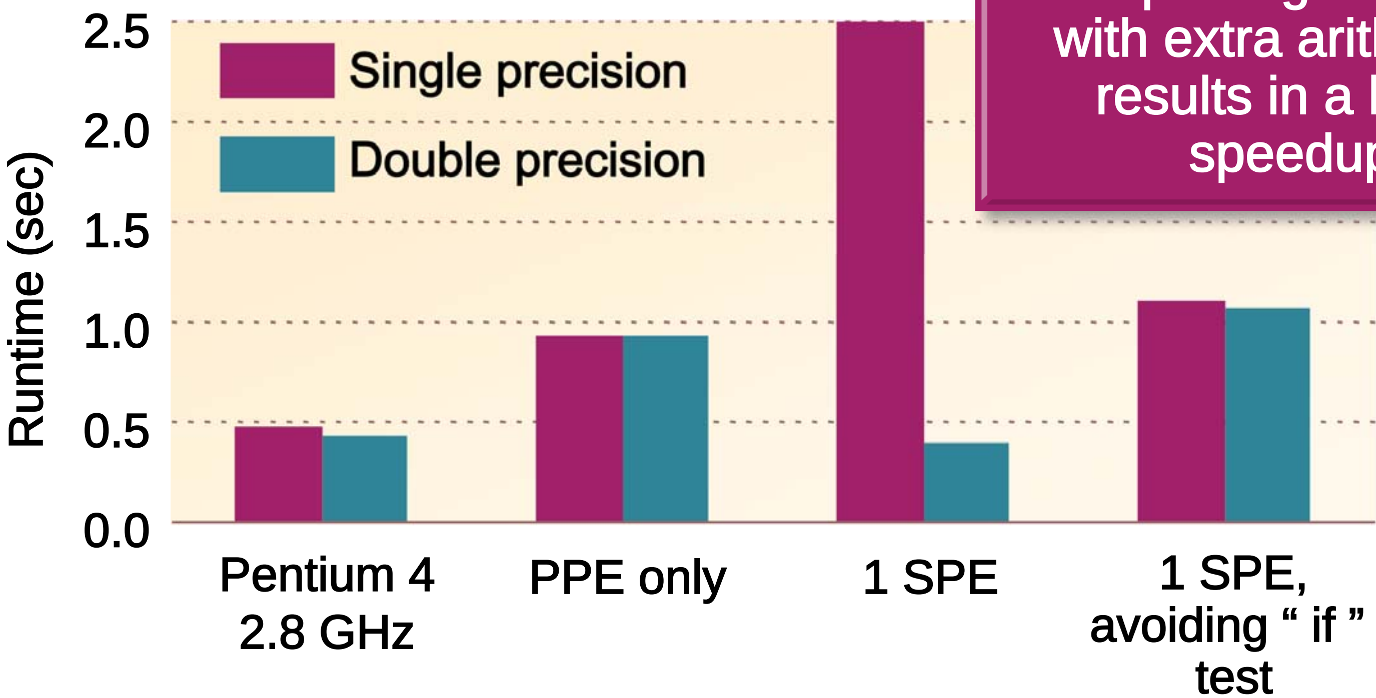
Cell broadband engine processor: Details

One 64-bit PPE	• Dual-threaded • Vector instructions (VMX)
Eight SPEs	• Dual-issue pipeline • Simple instruction set heavily focused on single instruction multiple data (SIMD) • Capability for double precision, but optimization for single • Uniform 128-bit 128-register file • 256-K fixed-latency local store • Memory flow controller with direct memory access (DMA) engine to access main memory or other SPE local stores

Genetic algorithm, traveling salesman: Single- vs. double-precision performance

- The cell processor has a much higher latency for double-precision results.
- The “ if ” test in the sorting predicate is highly penalized for double precision.

Replacing this test with extra arithmetic results in a large speedup

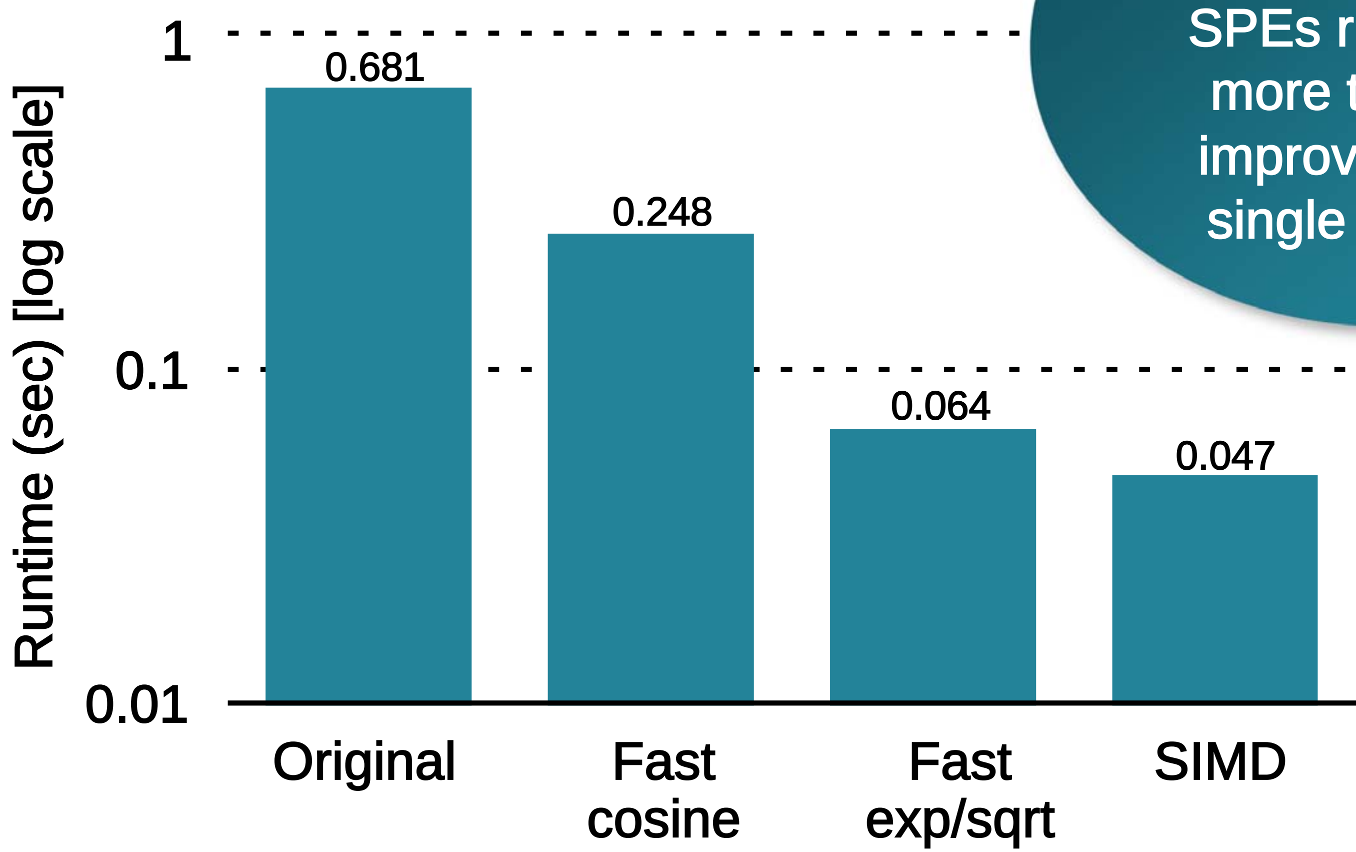


OAK RIDGE National Laboratory

Genetic algorithm, Ackley's function: Using SPE-optimized math libraries

Ackley's function involves cos, exp, sqrt

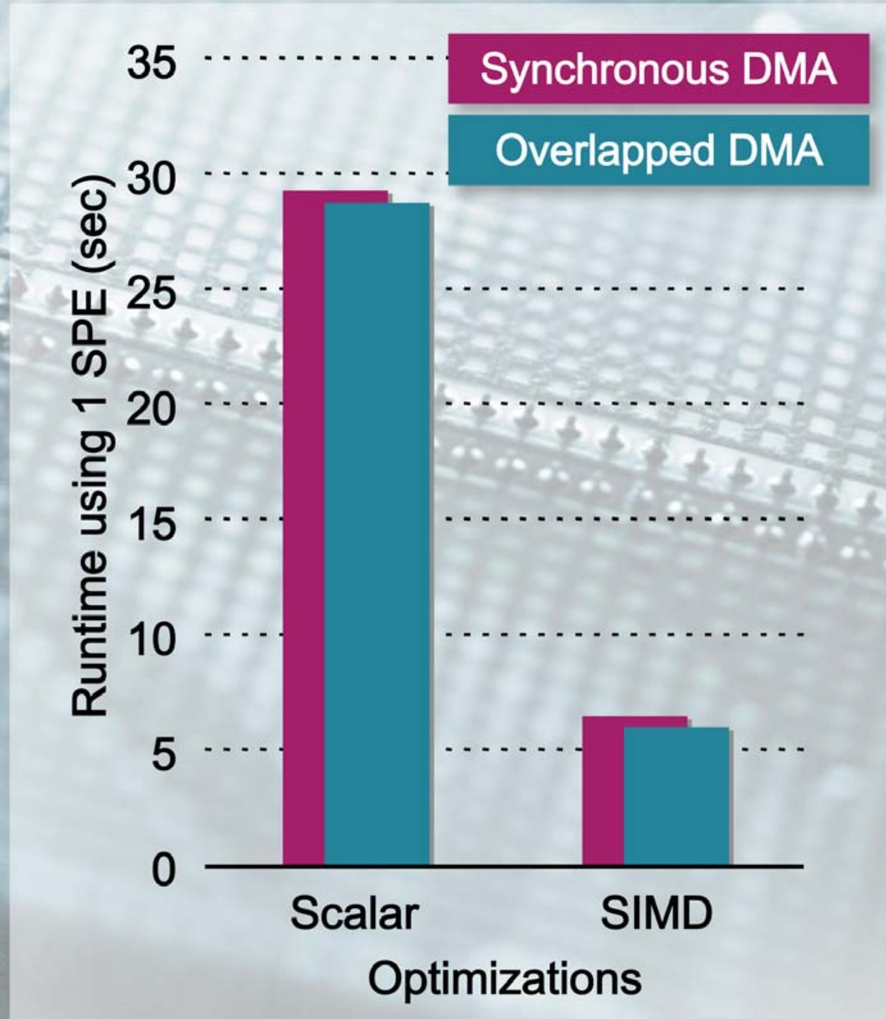
Switching to a math library optimized for SPEs results in a more than 10x improvement for single precision



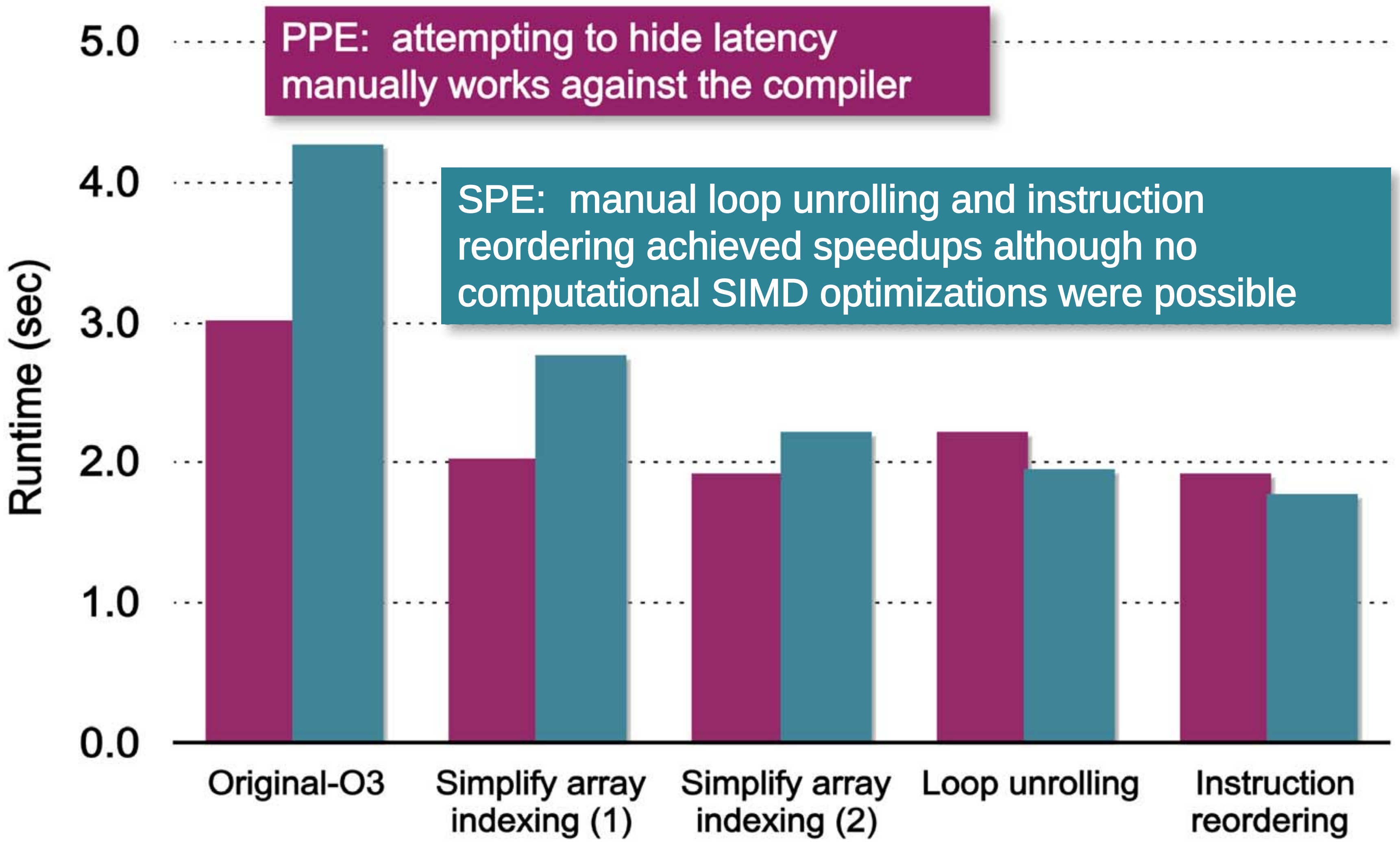
Oak Ridge National Laboratory

Covariance matrix creation: DMA communication overhead

- The cell processor can overlap communication with computation.
- Covariance matrix creation has a low ratio of communication to computation.
- However, even with an SIMD-optimized implementation, the high bandwidth of the cell's EIB makes this overhead negligible.

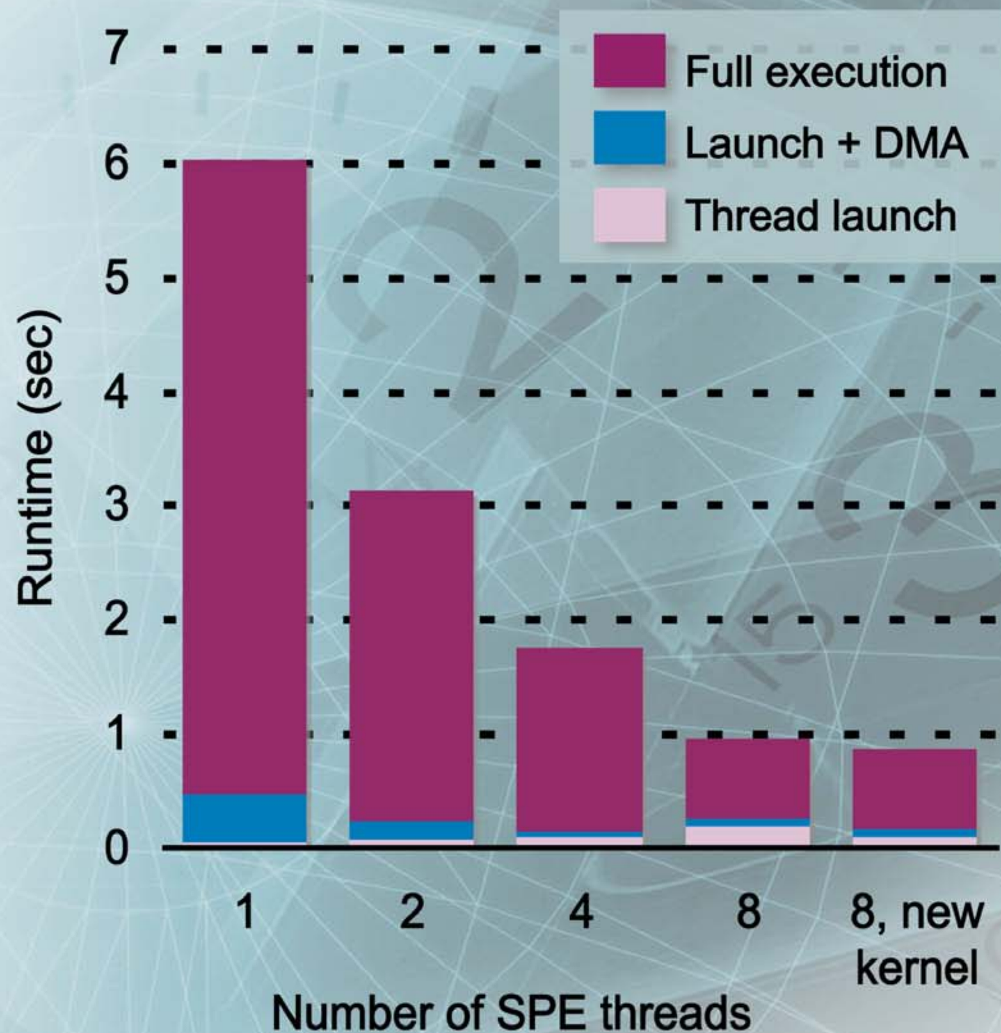


Stochastic Boolean SAT solver: Hiding latency in logic-intensive apps



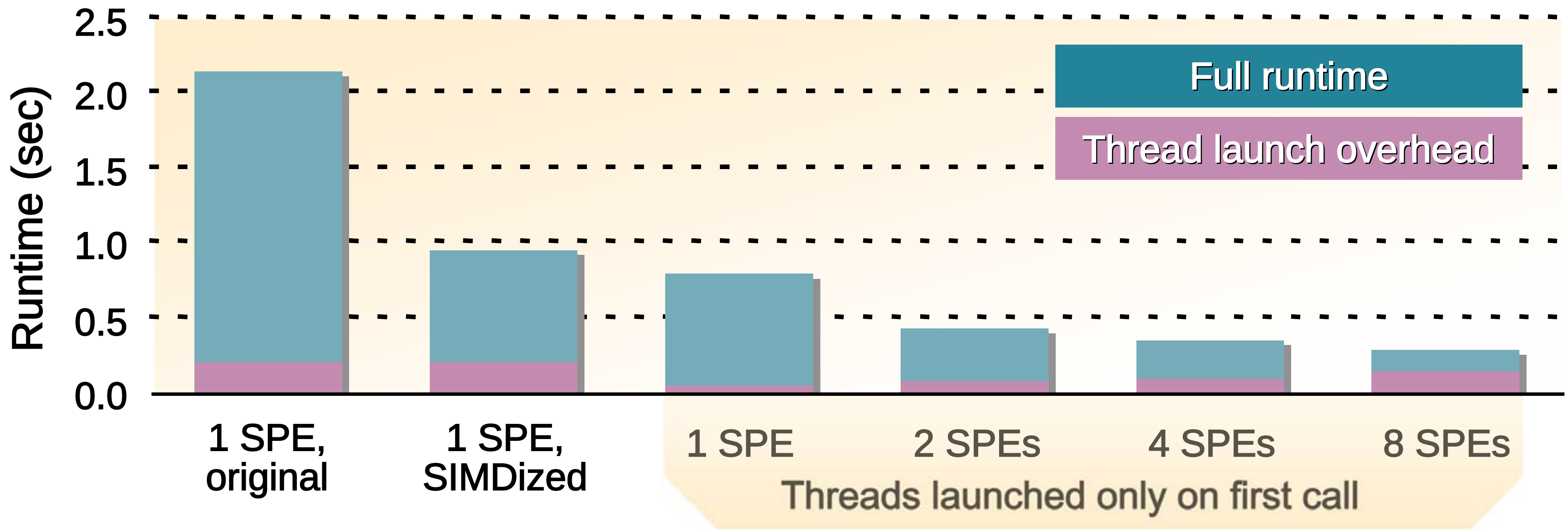
Support vector machine: Parallelism and concurrent bandwidth

- As more simultaneous SPE threads are added, the total runtime decreases.
- Total DMA time also decreases, showing that the concurrent bandwidth to all SPEs is higher than to any one SPE.
- Thread launch time increases, but the latest Linux kernel for the cell system does reduce this overhead.



Molecular dynamics: SIMD intrinsics and PPE-to-SPE signaling

- Using SIMD intrinsics easily achieved 2x speedups in total runtime.
- Using PPE-SPE mailboxes, SPE threads can be reused across iterations.
- Thus, thread launch overheads will be completely amortized on longer runs.



Oak Ridge National Laboratory

Conclusion

- Be aware of arithmetic costs.
  - Use the optimized math libraries from the SDK if it helps.
  - Double precision requires different kinds of optimizations.
- The cell has a very high bandwidth to the SPEs.
  - Use asynchronous DMA to overlap communication and computation for applications that are still bandwidth bound.
- Amortize expensive SPE thread launch overheads.
  - Launch once, and signal SPEs to start the next iteration.
- Use of SIMD intrinsics can result in large speedups.
  - Manual loop unrolling and instruction reordering can help even if no other SIMDization is possible.

Contact

Jeremy Meredith

Future Technologies Group
Computer Science and Mathematics Division
(865) 241-5842
jsmeredith@ornl.gov