
Experiences Programming the Cell Across a Diverse Set of Applications



Jeremy Meredith
Future Technologies Group

-
- ➔ Overview of the application kernels
 - Scientific, imaging, cognitive algorithms

 - ➔ Optimization strategies
 - “Asymmetric-Thread Runtime Model”
 - Parallelism, overheads, latencies, etc.

 - ➔ Performance results
 - 2.4GHz Cell
 - 2.2GHz Opteron

Application Kernels

- ➔ Monte Carlo Light Integration
- ➔ Molecular Dynamics
- ➔ Covariance Matrix Creation
- ➔ Boolean Satisfiability Solver
- ➔ Genetic Algorithms

Monte Carlo Light Propagation

- ➔ Simulation of point-source heating in an infinite isotropic scattering medium
 - (from Oregon Medical Laser Center)
- ➔ Fixed number of photons (outer loop)
- ➔ Variable number of steps per photon (inner loop)

Molecular Dynamics

- ➔ Force evaluation and integration between atoms
 - Lennard-Jones potential interaction model
 - Velocity Verlet integration algorithm
- ➔ Number of interacting atoms changes over time
- ➔ N^2 search over atom pairs for interacting atoms
 - Force evaluation over only those within cutoff limit
 - Search over atom pairs is bottleneck

Covariance Matrix Creation

➔ For each <a,b> entry in the LxL matrix Cov,

$$\text{Cov}_{a,b} = \sum_{i=1}^N \sum_{j=1}^M \text{input}_{i,j,a} \times \text{input}_{i,j,b}$$

- ➔ Applications include hyperspectral imaging
 - Can build concise model of background for subtraction from the HSI data cube
- ➔ Known loop counts, heavy data streaming, straightforward computation

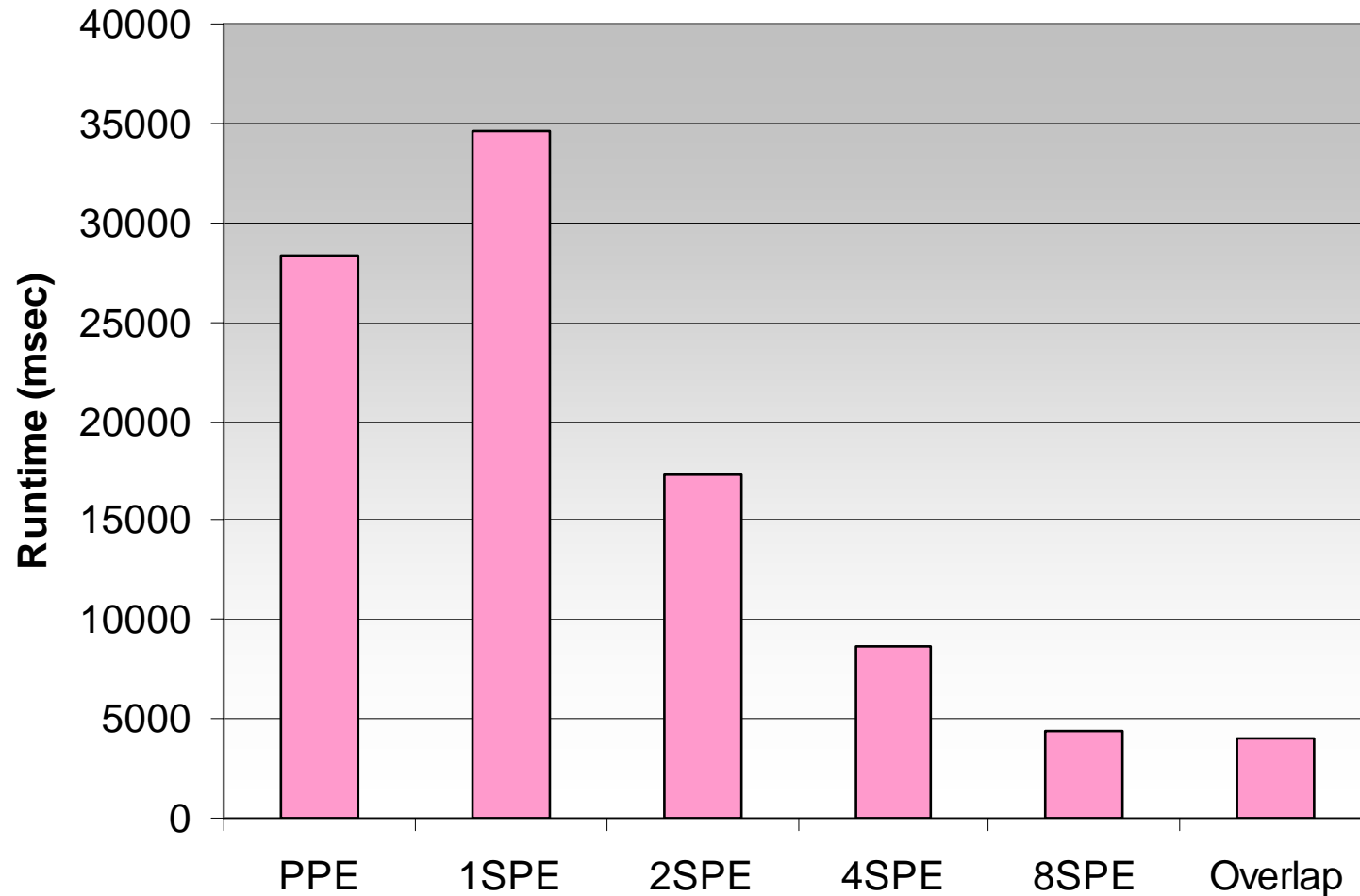
- ➔ Is there an assignment to a set of variables in a Boolean expression to make the entire expression true?
 - Many problems, like planning, can be reduced to SAT
- ➔ **Unit Propagation**
 - Stochastic solvers repeatedly change the value of a variable, updating the scores of clauses which refer to that variable
 - Main loop in solvers like GSAT, WalkSAT, HSAT
 - Inefficient to check all clauses; instead, update only clauses containing that variable
- ➔ **Essentially no computation involved**
 - lookup, read, modify, write of random memory locations

Genetic Algorithm

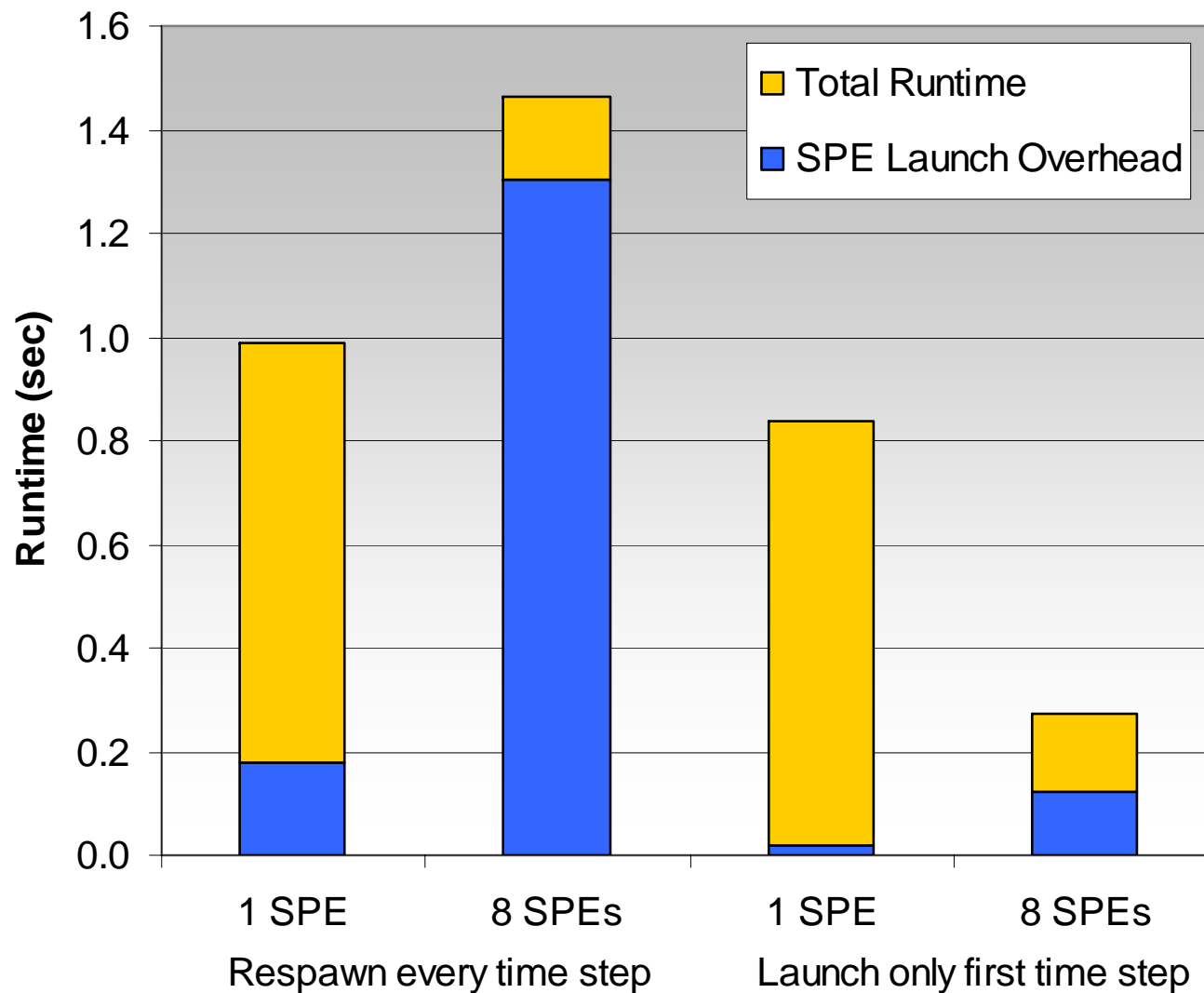
- ➔ Parallel optimization on a large populations
- ➔ Individuals selected for breeding by their fitness
 - replication, mutation, combination of “chromosomes”
- ➔ Fitness evaluation is typically bottleneck
- ➔ Two functions from GENE_s package:
 - Ackley’s function (more computation)
 - Traveling salesman (more logic, with sorting)

- ➔ Task-level parallelism
- ➔ SPE thread launch overhead
- ➔ SIMD optimizations
- ➔ Concurrent DMA bandwidth
- ➔ Overlapping communication and computation
- ➔ Latency hiding and loop unrolling
- ➔ SDK optimized math libraries
- ➔ Double precision penalties

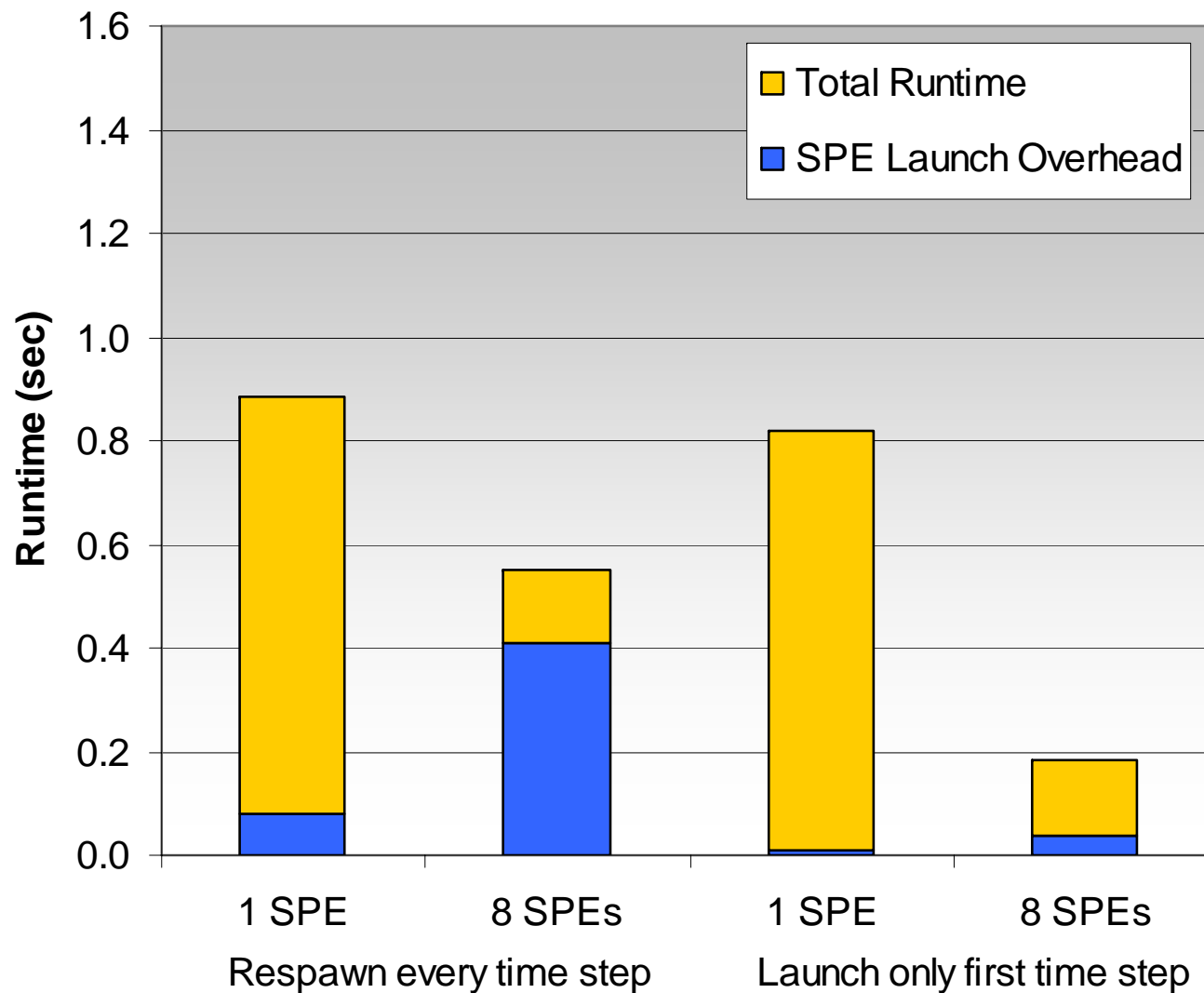
Monte Carlo Light Propagation: Task-level Parallelism



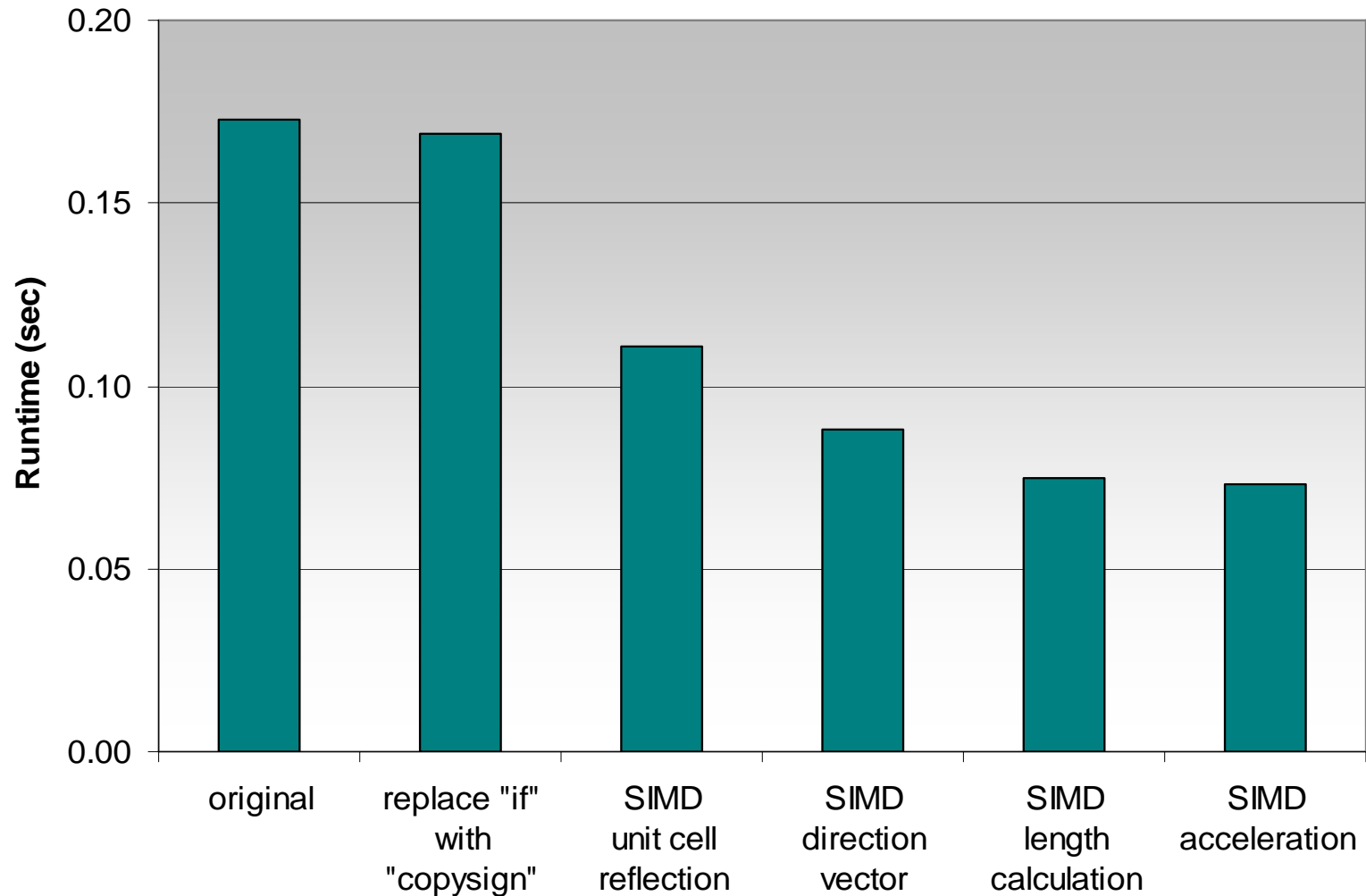
Molecular Dynamics: Thread Launch Overhead



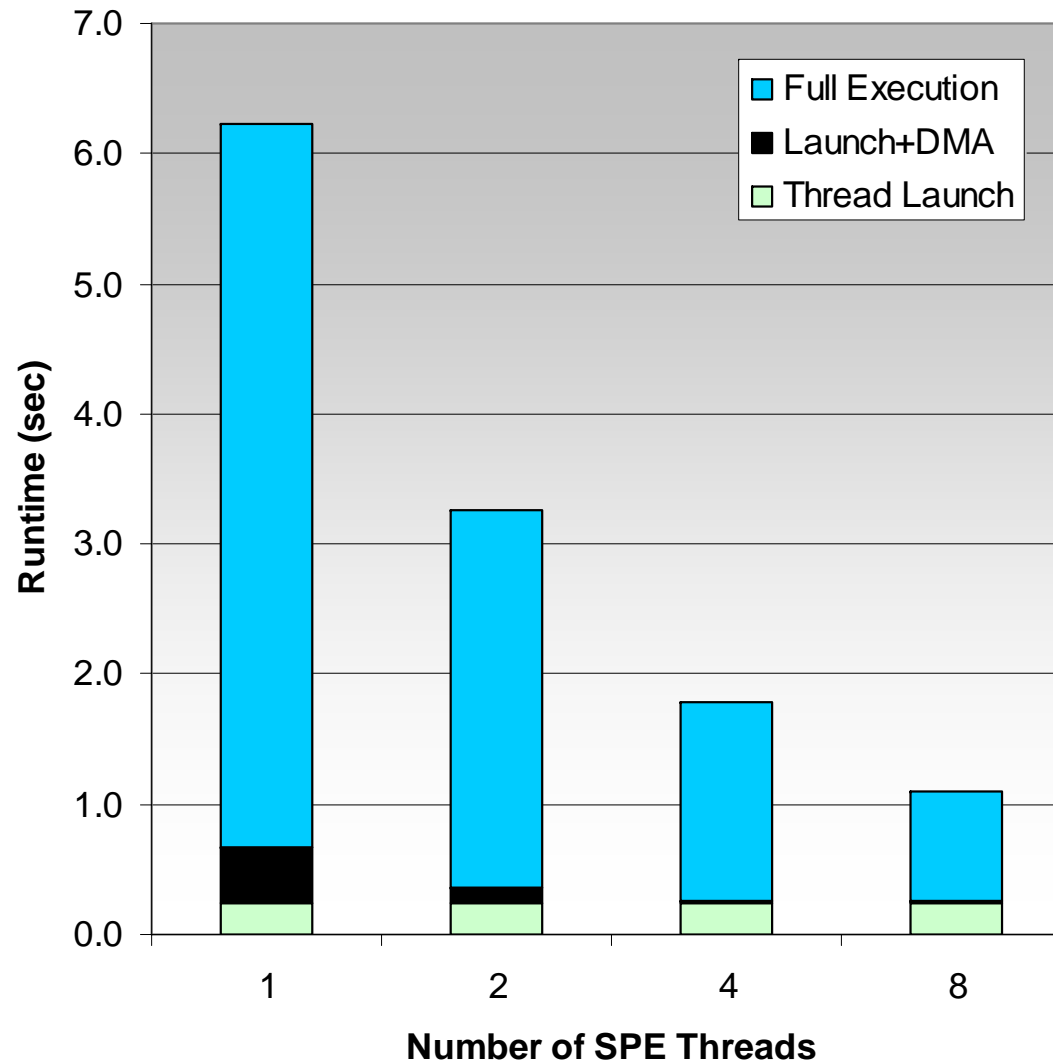
Molecular Dynamics: Thread Launch Overhead (cont.)



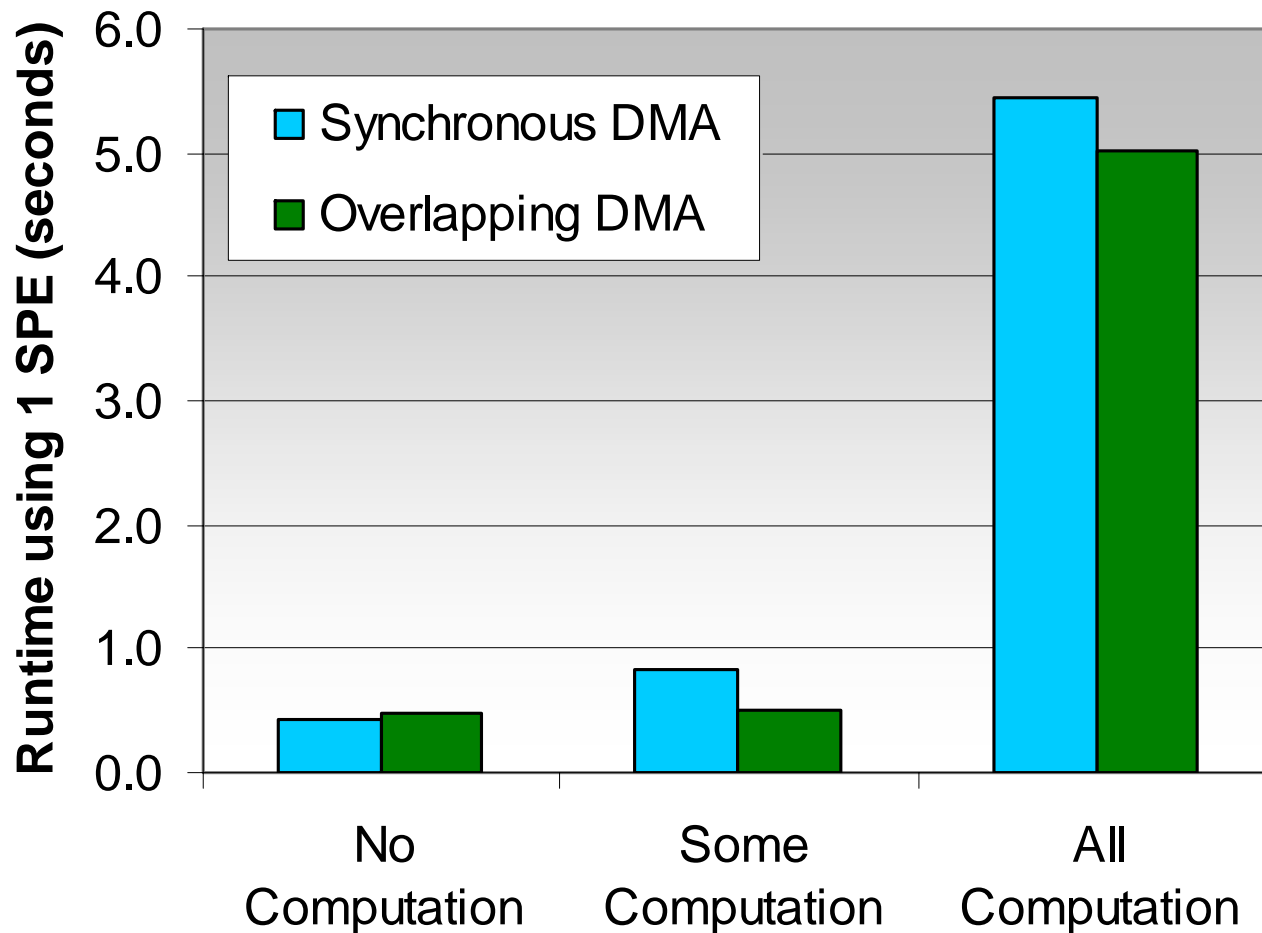
Molecular Dynamics: SIMD Optimization



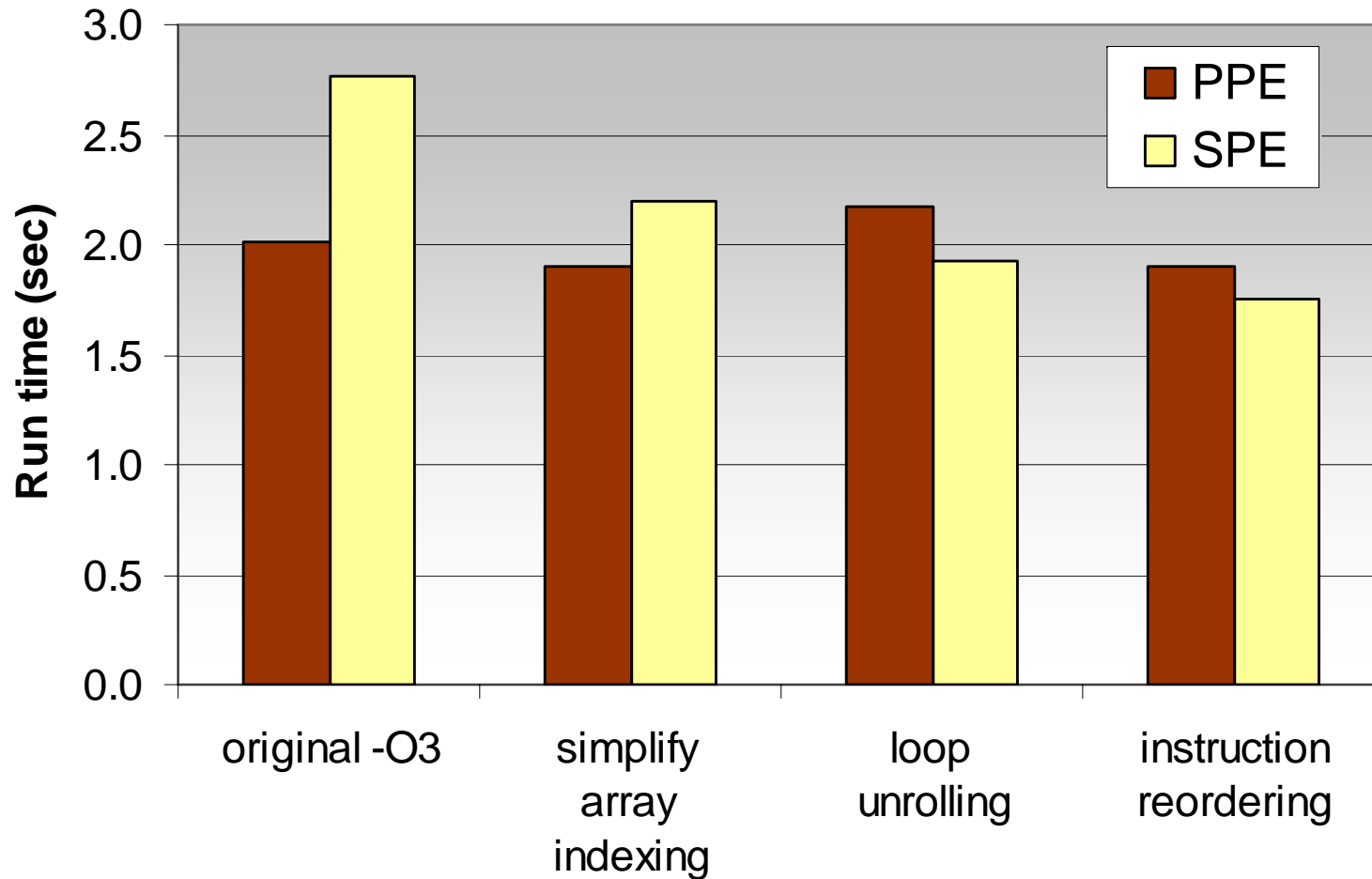
Covariance Matrix Creation: Concurrent Bandwidth



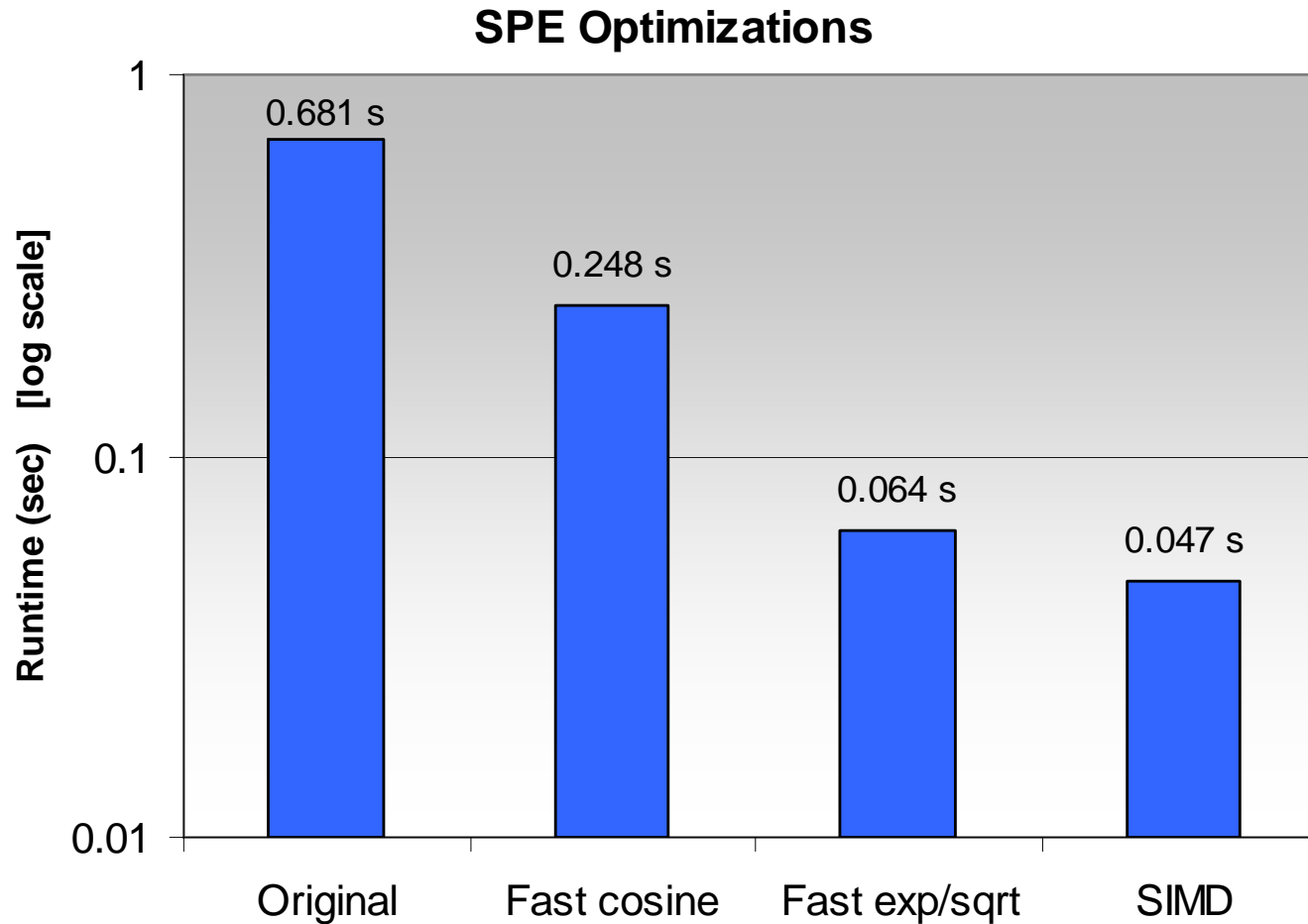
Covariance Matrix Creation: Asynchronous DMA Communication



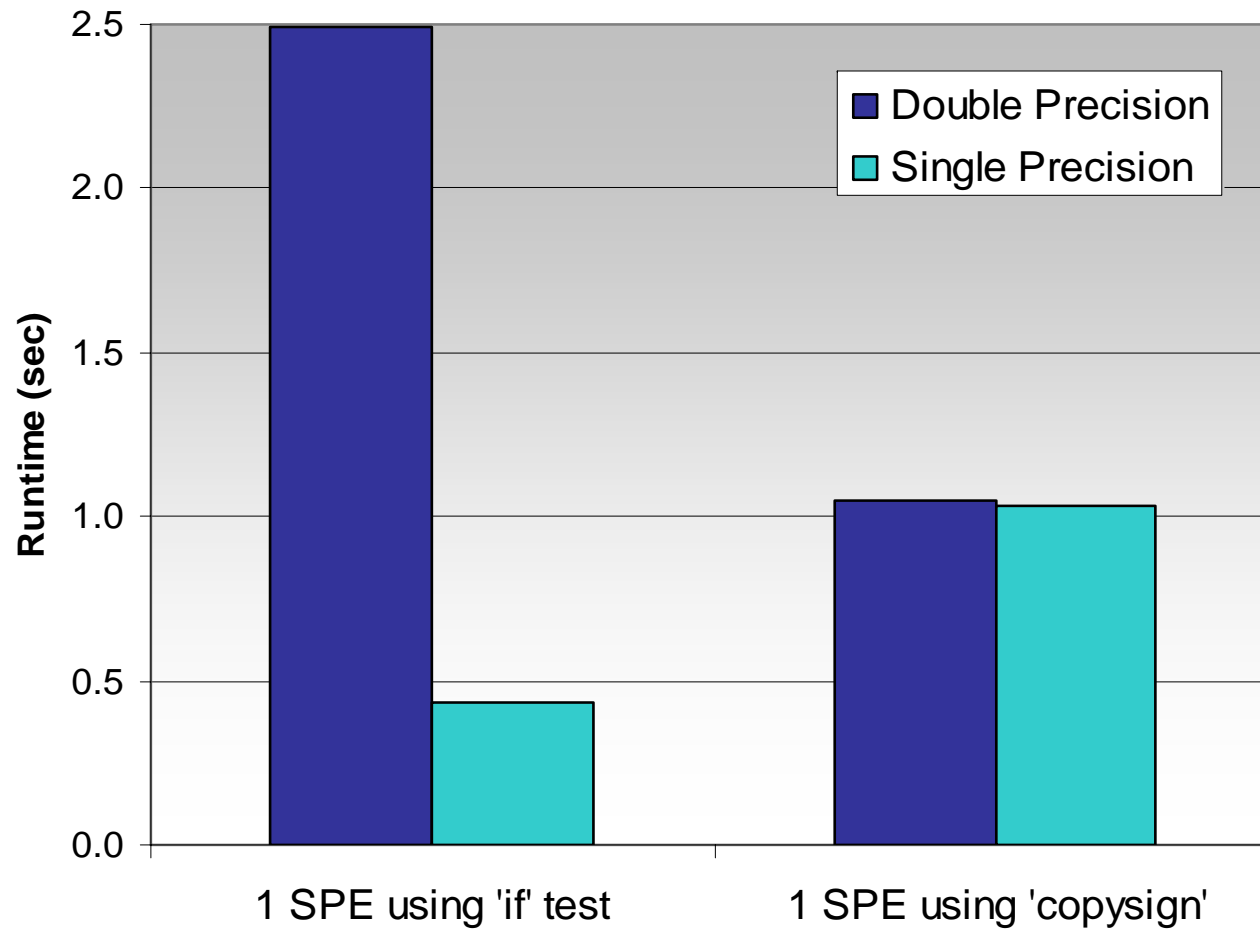
Boolean Satisfiability (SAT) Solver: Latency Hiding and Loop Unrolling



Genetic Algorithm, Ackley's Function: SDK Optimized Math Libraries



Genetic Algorithm, Traveling Salesman: Double Precision Penalties



Performance Results

- ➔ Monte Carlo Light Integration
- ➔ Molecular Dynamics
- ➔ Covariance Matrix Creation
- ➔ Boolean Satisfiability Solver
- ➔ Genetic Algorithms

Monte Carlo Light Propagation

- ➔ Good scaling across all 8 SPEs can't help the variable length, short inner loop, almost no SIMD, and heavy reliance on random number generation

# Photons	1000	10000	100000
Opteron	24 ms	232 ms	2384 ms
Cell, 8 SPEs	38 ms	357 ms	3112 ms
Cell, PPE only	288 ms	2843 ms	28384 ms

Molecular Dynamics

- ➔ A fair amount of SIMDizable computation lets even a single SPE beat the Opteron.
- ➔ All 8 SPEs are about 5x faster than the Opteron.

# Atoms	512
Opteron	0.925 sec
Cell, 1 SPE	0.816 sec
Cell, 8 SPEs	0.181 sec
Cell, PPE only	4.701 sec

Covariance Matrix Creation

- ➔ High concurrent bandwidth and straightforward computation allows efficient use of all 8 SPEs.
- ➔ 8 SPEs are almost 20x faster than the Opteron.

Data Set Size	256x65k
Opteron	12.308 sec
Cell, 8 SPEs	0.662 sec
Cell, PPE only	88.290 sec



Boolean Satisfiability (SAT) Solver



- ➔ No computation, indirect lookups let a single SPE only barely beat even the PPE
- ➔ A single SPE is 3.4x slower than the Opteron
 - However, multiple SPEs could search independent parts of the problem space

# Vars	800
# Flips	10M
Opteron	0.571 sec
Cell, 1 SPE	1.961 sec
Cell, PPE Only	1.998 sec

Genetic Algorithm

- ➔ Compute-intensive Ackley's function:
one SPE is 4x faster; eight SPEs are 21x faster

Population Size	262k	1.05M
Opteron	0.645 sec	2.514 sec
Cell, 1 SPE	0.165 sec	0.637 sec
Cell, 8 SPEs	0.060 sec	0.119 sec
Cell, PPE only	2.797 sec	11.146 sec

- ➔ Logic-intensive traveling salesman:
one SPE is 4x slower; eight SPEs are 2x faster

Population Size	131k	524k
Opteron	0.466 sec	1.876 sec
Cell, 1 SPE	1.697 sec	6.761 sec
Cell, 8 SPEs	0.248 sec	0.884 sec
Cell, PPE only	3.802 sec	15.209 sec

Optimization Strategy Summary

- ➔ Parallelization across the SPEs is critical
- ➔ Be aware of arithmetic costs
 - Use the optimized math libraries from the SDK if it helps
 - Double precision requires different kinds of optimizations
- ➔ The EIB has a very high bandwidth to the SPEs
 - Use asynchronous DMA to overlap communication and computation for apps with heavy bandwidth needs
 - But for many apps, it may simply waste space in the SPE LS
- ➔ 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



Acknowledgements and More Info



- ➔ This research was sponsored by the Office of Mathematical, Information, and Computational Sciences, Office of Science, U.S. Department of Energy under Contract No. DE-AC05-00OR22725 with UT-Battelle, LLC. Accordingly, the U.S. Government retains a non-exclusive, royalty-free license to publish or reproduce the published form of this contribution, or allow others to do so, for U.S. Government purposes.
- ➔ <http://www.csm.ornl.gov/ft>
- ➔ jsmeredith@ornl.gov