SUPER Performance Auto-Tuning and MPAS-Ocean Optimization

### MFDn Application (NUCLEI)
- Many Fermion Dynamics: Nuclear (MFDn) is used to calculate the properties of light atomic nuclei.
- Nominally, this requires an eigensolver like Lanczos which requires applying the operator repeatedly.
- MFDn forms the very large (half a billion nonzeros per process) symmetric configuration interaction matrix explicitly.
- As such, repeated application of the operator is particularly bandwidth-intensive and time-consuming.
- Symmetry further complicates this as it necessitates high performance SpMV and SpMV_T (transpose)

### LOBPCG
- LOBPCG is a block eigensolver that restructures computation into a series of Sparse Matrix-Dense Matrix Multiplications (SpMM) where the dense matrix is tall and skinny (500,000 x 16 per process).
- Once again, symmetric necessitates SpMM_T (transpose)
- The advantage (assuming comparable convergence) is that one can reduce the number of times the matrix must be read by a factor of m (the number of vectors in block). Ideally, this should result in a m-fold increase in performance.
- Unfortunately, naive implementations of SpMM and SpMM_T failed to deliver the expected performance.
- SUPER collaborated with FastMath to model, analyze, and optimize these operations.

### Compressed Sparse Blocks (CSB)
- Using the CSB data format, we tiled each local matrix into BxB tiles.
- Through the use of CSB and tuning of B, we were able to improve performance by 1.5x for SpMM and 3x for SpMM_T.
- Unfortunately, performance seemed to saturate at m~12.
- We constructed a series of Roofline Models to capture the effects of limited L2 and L3 locality on finite L2, L3, and DRAM bandwidth.
- The result clearly shows when one can no longer maintain a working set of vector data in the L2 a transition from DRAM-limited to L3-limited performance

---

### MPAS-Ocean Application
- The Model for Prediction Across Scales (MPAS) is a modeling framework for climate simulations.
- MPAS utilizes a fully unstructured mesh, which allows more flexibility in the description of the mesh locations.
- But it also has negative impact on performance due to multiple factors such as unordered data elements, irregular memory access patterns, mesh partitioning and communication imbalance.
- Regression analysis of scaling experiments shows scalability bottlenecks with runtime functions of the form $T_p = c_1 + c_2 \times (T_p/p)$, where $c_2$ is a large constant, thus limiting scalability; investigating the reasons for this behavior.

### MPAS-Ocean Optimizations
1. To improve compiler-based vectorization of MPAS-Ocean, the data structures were rewritten utilizing hash tables. 10-15% performance improvement was achieved.
2. Data element reordering based on Space Filling Curves was done to improve on-node cache performance. Up to 20% reduction in overall application runtime was observed.
3. On-node threading using OpenMP was explored showing significant reduction in number of FLOP per core.
4. A benchmark for reorganization of halo-exchanges through data aggregation was written. Benchmark shows significant potential in communication performance improvement.
5. A new weighted graph partitioning scheme “Hindsight” was developed. About 10-15% reduction in overall runtime was observed.
6. A new weighted hypergraph-based and halo-aware partitioning scheme was developed. About 20% reduction in overall runtime was observed.

---

### Autotuning Tool Integration
- Tools must support architectural changes and application requirements.
- Tools are extended in response to application tuning.

### Example Extension: Sparse Matrices
#### GOAL: Extend compiler transformations and code generation to non-affine bounds and subscripts with index array. Compose with other transformations
- Performance comparison with manually-written CUSP library
  - Using autotuning, outperforms CUSP library
  - CSR Vector: parallelize by nonzero

---

### Example Tool for MPAS
#### Performance Issue:
Significant structure indirection inside loop nests block/5mesh/iceellsOnEdgeArray/L,Edge
**Solution:** Replace with pointer buffers that point directly to array

---

### Autotuning Experiments for MPAS
- **Conclusion:**
  - Partition on icell, but value unknown before partitioning

---

M. Hall, K. Huck, J. Hollingsworth, L. Oliker, A. Malony, S. Moore, A. Sarje, S. Shende, S. Song, S. Williams, P. Worley

*Support for this work was provided through the Scientific Discovery through Advanced Computing (SciDAC) program funded by the U.S. Department of Energy Office of Advanced Scientific Computing Research*