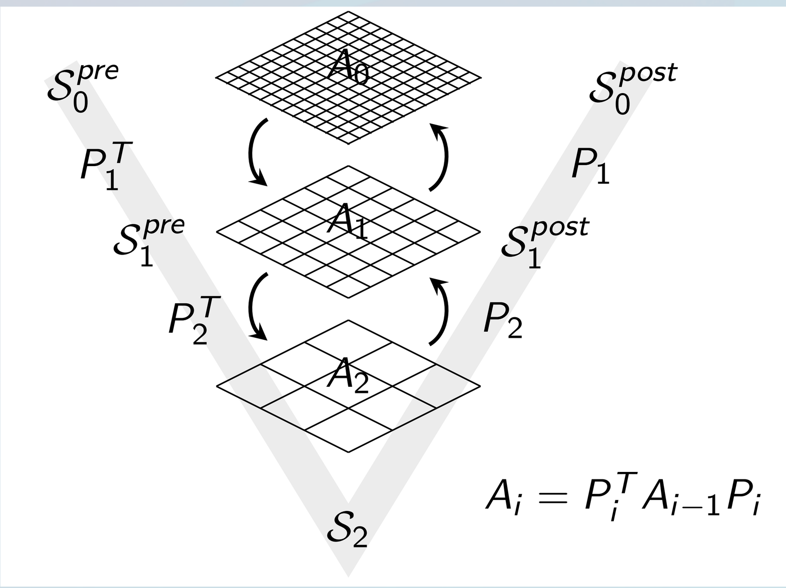
SAND2016-5179O

Reducing Communication and Computation in Scientific Computing

Sandia National Laboratories
Grey Ballard, Truman Fellow
Sandia National Laboratories, California 94551

Problem

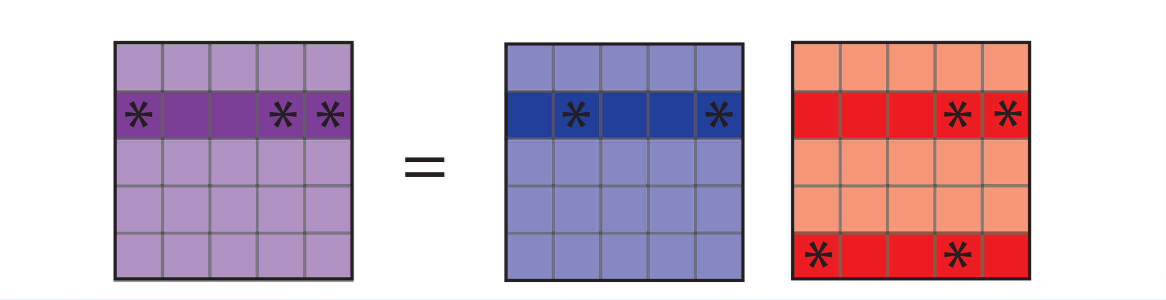
- Moving data is more expensive than computing with it, and the problem is getting worse
- Data-intensive tasks are becoming more of a bottleneck for scientific computing applications
- The overall goal of this Truman project is to address this problem for several fundamental computations within scientific computing
- This poster focuses on one particular problem: sparse matrix-matrix multiplication (SpGEMM) in the setup phase of algebraic multigrid (AMG)



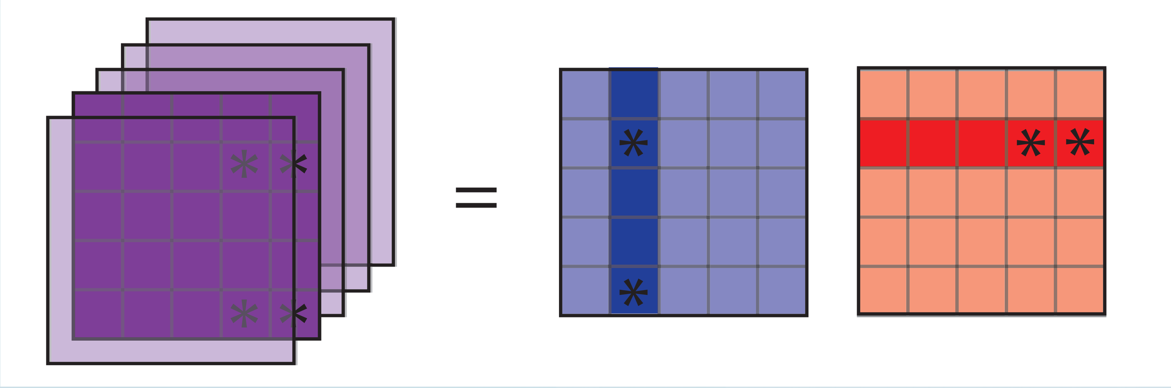
Approach

- AMG setup is bottlenecked by SpGEMM, SpGEMM is bottlenecked by interprocessor communication
- Kernel computation is triple product: $A_i = P_i^T A_{i-1} P_i$
- Key idea is to choose the correct algorithm for each SpGEMM to minimize communication

Row-wise Algorithm



Outer-Product Algorithm



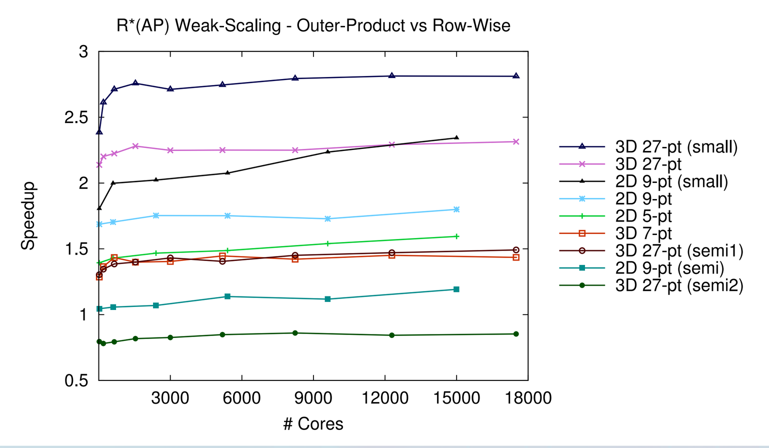
Results

- Theoretical analysis of model 2D/3D problems concludes that row-wise algorithm should be used for *AP* and outer-product algorithm should be used for $P^T(AP)$

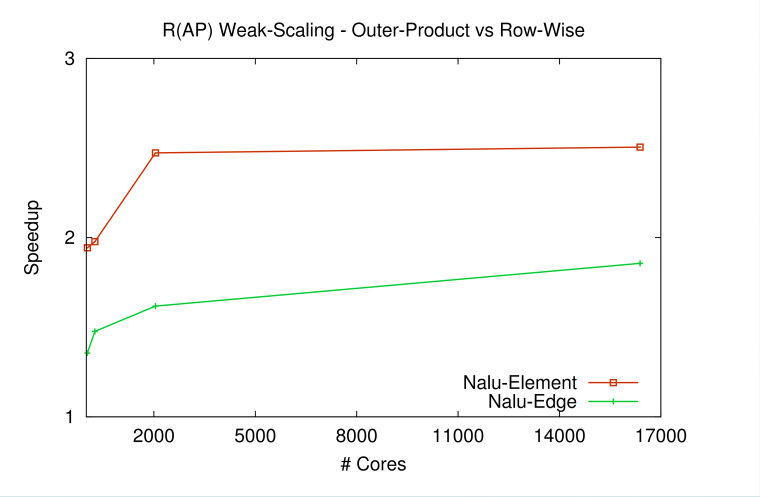
	Row-Wise	**Outer-Product**	**Ratio** 2D	3D
$AP = A_f \cdot P$	$2(5/3)^{D-1}$	$2(7/3)^{D-1}$	0.7	0.5
$A_c = P^T \cdot (AP)$	$2(7/3)^{D-1}$	2	**2.3**	**5.4**

Table entry gives maximum per-processor communication cost divided by the size of the processor's fine grid halo

- Experimental analysis of model problems (using Trilinos) confirms superiority of outer-product algorithm for $P^T(AP)$, up to 2.8x speedup



- Experimental analysis of application problems (from Nalu) demonstrates up to 2.5x speedup



Significance

- New algorithm for SpGEMM implemented in Trilinos and Muelu for algebraic multigrid preconditioning
- Using outer-product algorithm for $P^T(AP)$ resulted in 16% improvement in overall setup for Nalu problem
- 57 citations for 14 Truman project publications, including 1 best paper award and 1 survey article
- Communication-efficient algorithms demonstrated for other computations, including dense matrix multiplication, nonnegative matrix factorization, and Tucker tensor decomposition

1. Ballard G, Druinsky A, Knight N and Schwartz O (2016), "Hypergraph Partitioning for Sparse Matrix-Matrix Multiplication", March, 2016. (arXiv 1603.05627). Submitted to TOPC.
2. Azad A, Ballard G, Buluc A, Demmel J, Grigori L, Schwartz O, Toledo S and Williams S (2015), "Exploiting Multiple Levels of Parallelism in Sparse Matrix-Matrix Multiplication", 2015. (arXiv 1510.00844). SISC, under revision.
3. Ballard G, Druinsky A, Knight N and Schwartz O (2015), "Brief Announcement: Hypergraph Partitioning for Parallel Sparse Matrix-Matrix Multiplication", In SPAA, 2015. pp. 86-88. ACM.
4. Ballard G, Siefert C and Hu J (2015), "Reducing Communication Costs for Sparse Matrix Multiplication within Algebraic Multigrid", September, 2015. (SAND2015-3275). To appear in SISC.