# Porting Hypre's BoomerAMG to Heterogeneous Computer Architectures: Strategies, Experiences and Optimizations



ECP COMMUNITY BOF: Recent Advances in Selected Sparse Linear Solvers Libraries – Virtual Meeting

May 10, 2022



#### LLNL-PRES-831941

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract DE-AC52-07NA27344. Lawrence Livermore National Security, LLC



Ruipeng Li



- **Sierra** at LLNL, **Summit** at ORNL. NVIDIA V100
- **Perlmutter** at NERSC. NVIDIA A100
- **Frontier** at ORNL. AMD Instinct 250X
- **El Capitan** at LLNL, AMD Instinct GPUs
- **Aurora** at ANL, Intel GPUs
- All have "Fat node" architecture. Most of the FLOPS are on the GPUs





### Complete AMG setup and solve on GPUs

- $\blacktriangleright$  AMG setup
  - Coarsening: PMIS
  - Interpolation: direct, MM-ext;
  - Aggressive coarsening; Multipass interpolation
- $\blacktriangleright$  AMG solve
  - $\blacksquare$  Matrix-by-vector: save local  ${\pmb P}^T$
  - Smoother: Jacobi; Gauss-Seidel; Chebyshev and GMRES polynomials (NEW!)

+628ma	s 648ma	666ma	1660m	+ 700mm		+720ma	+ 740ms	+ 760ma	+380ma	+80
www.www.www	in inner	18	an an isana		1.00	and the second	A 2018 damage	and a state of the	-	
11001							1.1.1	1 11 million		
11000								1 1 1 <b>-</b> 1 1		
1110001111011			in the same	1.1	11 1	2022 21				
12108.112			1. 19 200	Rasa		2000	1.1.200	a		
	1 Server 0 [152 378 /rei]	1	i Dawi	Ko 14 1962 Leven 1 (* 156			A DE DATE DE LA COMPANY	2	A TEACH	i i
AND Cathering (M.	1 Lavier (* 1533) (* 1) 233 mai Tupin (*	1 AAAP (25.5564 mm)		PCG 10 AMG Leven 1 (41:156 CPErtary (21.34) res	nea (rea) (rea)		APA na 11 Addition	2	200 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2 - 2	2 (2008 8 2 (20) 8 2









i s l li li li lindon melli i lin		1.0	1				
AMG Level 0 [63.278 ms]							
Creatin PMIS (7.015 ms)	ExtPlinterp [28.233 ms]		TripleMat-RAP [26.004 ms]				
Extract Submatrix [5.8	Comput Matrix-matrix mult [7.799 ms] Ex Truncation	(9.153 ms) CSRMatrix	Aultiply [ CSRMatrixM				
	Stream5	Street	imSync0 StreamS				
	Stream						
Countries of the International Countries of the International State	11 CudaStr	thronize) ( cu) ( ) 👢 🚛 👘 🔤 🚛 ( c) ( cuda	StreamS)				
	🐘 🛑 🖓 cudaStr) 🔰 👘 🛑) cudaStr) 🚺 🗍 👘 cudaStreamSyn	chronize) cu [ ] / / for any c cudz	StreamS cudaStr cudaStreamSynchronize				







**1** node: CPU: 64 MPI, GPU: 8 GPUs,  $n \times n \times n$  grid



- $\blacksquare$  \_\_\_\_: default + Jacobi (0.85)
- ....: 2s: +1 lvl aggr. coarsening, 2-stage interp.
- .....: mp: +1 lvl aggr. coarsening, multipass interp.





 $\blacksquare$  Coarse-grid operator:  $\boldsymbol{P}^\mathsf{T} \boldsymbol{A} \boldsymbol{P}$ 

2 Aggressive coarsening: 2nd stage coarsening  $S_{CC}^{(A)} = (S^2 + S)_{CC}$ 

- **3** MM-ext interpolation
  - PMIS requires *extended* interp. (De Sterck 2008) with distance-2 interp. sets
  - Original formulation is difficult for efficient implementation on GPUs
  - New formulation in matrix-matrix (MM) multiplications

$$W = - ig[ (D_{FF} + D_{\gamma})^{-1} (A^s_{FF} + D_{eta}) ig] ig[ D^{-1}_{eta} A^s_{FC} ig] \equiv - ilde{A}^s_{FF} ilde{A}^s_{FC},$$

**4** Multipass interpolation in MM forms







• MM-ext: 
$$W = -(D_{FF} + D_{\gamma})^{-1} (A_{FF}^s D_{\beta}^{-1} + I) A_{FC}^s \equiv -Y A_{FC}^s$$

$$\blacksquare A = D + A^s + A^w, D_\beta = \operatorname{diag}(A_{FC}\mathbf{1}_c), D_\gamma = \operatorname{diag}(A_{FF}^w\mathbf{1}_f + A_{FC}^w\mathbf{1}_c)$$

• Easy to verify 
$$-YA_{FC}^s \mathbf{1}_c = \mathbf{1}_f$$

• Assuming 
$$A^w = 0$$
 and  $D_\beta = -\text{diag}(A_{FF}1_f) = -D_\alpha$ 

$$egin{aligned} I + YA_{FF} &= I + D_{FF}^{-1}((A_{FF} - D_{FF})D_{lpha}^{-1} - I)A_{FF} \ &= (I - D_{FF}^{-1}A_{FF})(I - D_{lpha}^{-1}A_{FF}) \end{aligned}$$

•  $I + YA_{FF} \approx 0$  if  $A_{FF}$  is strongly d.d, i.e.,  $Y \approx A_{FF}^{-1}$ 







- Hash-based *sparse accumulators*; rows are *unsorted*
- Naive row NNZ bound: number of intermediate products
- symbolic multiplication: row NNZ count (or bound)
- *numeric* multiplication with adequate memory allocated
- Stochastic estimator (Cohen, 1997): good estimate
  - standard deviation  $\sigma = 1/\sqrt{r-2}$ .  $\hat{z}' = \frac{\hat{z}}{1-3\sigma}$
  - $\hat{z}' \ge z$  and  $\hat{z}' \le \frac{1+3\sigma}{1-3\sigma}z$  with high probabilities







Squaring 125-point matrix



8/15



## Optimizing sparse matrix-matrix kernel (cont'd)

- Robust hash probings
  - 1 linear:  $(h'(k) + i) \mod s$
  - **2** quadratic:  $(h'(k) + c_1i + c_2i^2) \mod s;$
  - **3** double:  $(h'(k) + ih_2(k)) \mod s$
- Load balancing: binning with row NNZ
  - $\begin{array}{l} & \langle \texttt{SHMEM\_HASH\_SIZE}, \texttt{GROUP\_SIZE} \rangle = \\ & \langle 32, 2 \rangle \dots \langle 16384, 1024 \rangle, \langle 16, 2 \rangle \dots \langle 8192, 1024 \rangle \end{array}$
- Load balancing: thread-group partition



Triple matrix product kernel











# GPU utilization deteriorates with smaller problems and coarser levels

Known issues that lower GPU utilization

- Synchronizations: Thrust syncs stream after calls
  - (*temporarily*) replaced with custom kernels
- **2** D2H and H2D copies
- 3 Small kernels

4 MPI

- GPU utilization (single GPU)
- "CPU overhead" = kernel launch overhead + CPU computation

	100x100x1	.00	100x100x2	200	200x200x200		
	setup	solve	setup	solve	setup	solve	
time	49.7	38.1	79.3	60.7	252.9	186.5	
GPU time	36.49	36.6	65.3	58.8	239.4	185.5	
CPU overhead	26.58%	3.94%	17.65%	3.13%	5.34%	0.54%	

#### ▶ For the "really small" 100<sup>3</sup> problem,

		-					
100x100x100 setup	mxl2	mxl3	mxl4	mxl5	mxl6	mxl7	mxl8
time	19.6	31.9	37.6	41.5	44.8	47.6	49.7
GPU time	17.9	28.6	31.9	33.8	35.1	35.7	36.49
CPU overhead	8.67%	10.34%	15.16%	18.55%	21.65%	25.00%	26.58%





## Overlapping GPU execution with CUDA-aware MPI

- Pack halo data
- StreamSync
- MPI-Isend/Irecv
- $\blacksquare$  local matvec
- MPI-waitall
- Pack halo data
- StreamSync
- $\blacksquare$  local matvec
- MPI-Isend/Irecv
- MPI-waitall







11/15



#### $\blacktriangleright$ Again, kernels are too small









### Putting them together: more optimized AMG on GPUs



■ The work of optimized SpGEMM was not included in "opt"







- ▶ hypre's structured solvers, AMG solvers, and more have been ported to heterogeneous GPUs
  - CUDA/HIP/oneAPI libraries  $\Rightarrow$  Thrust/rocPRIM/oneDPL  $\Rightarrow$  kernels
- > AMG algorithms on GPUs have been "stabilized", and are under optimizations
- $\checkmark$  Port BoomerAMG to Intel GPUs
- $\checkmark$  More coarsening algorithms on GPUs other than PMIS
- $\checkmark$  New Semi-structured solvers on GPUs

R. Falgout, R. Li, B. Sjogreen, U.M. Yang, and L. Wang, "Porting hypre to Heterogeneous Computer Architectures: Strategies and Experiences", Parallel Computing, 2021







Center for Applied Scientific Computing

#### THANK YOU!

Questions & Comments

li50@llnl.gov

#### Disclaimer

This document was prepared as an account of work posmored by an agency of the United States government. Neither the United States government to Larence Livensen National Scority (L.C. en au of their employees makes any variant), expressed or implied, or assumes any logal holdby or responsibility for the accuracy, completeness, or worklasse of any information, apparatus product, or process disclosed, or represents that is use would not infining privately constrained fractional terms on the present commercial product, process, or service by trade name, trademark, namefacturer, or otherwise does not necessarily constitute or highly its indexemment, measurement on the forcing by the United States government or Lavernee Livence National Scority, (L.C. The viscos and opinion of androne segments priority does not accessarily state or reflect theor of the United States government or Lavernee Livence National Scority, (L.G. and kall not be used for advertising or product redocencem propose.

