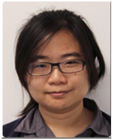
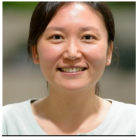
# PIRT
# Parallel Iterative Reconstruction for Tomography with error correction

Team

Sajid Ali
(Northwestern Univ.)



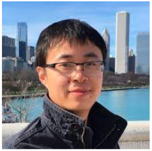
Panpan Huang
(Northwestern Univ.)



Wendy Di
(Argonne Natl. Lab.)

Mentors

Barry Smith
(Argonne Natl. Lab.)



Hong Zhang
(Argonne Natl. Lab.)

# Optimization formulation

## Least squares cost function for tomography inversion, with error correction

- To recover both shifts and object :
$\underset{\mathcal{W}\geq 0, P_\theta}{min} \phi(\mathcal{W}, P_\theta) = \frac{1}{2}||\mathcal{L}\mathcal{W} - g(\mathcal{D}, P_\theta)||$
- First order derivatives analytically computable :
$\nabla\phi(\mathcal{W}, P_\theta) = [\mathcal{L}^T, \nabla_{P_\theta}\phi(\mathcal{W}, P_\theta)]^T(\mathcal{L}\mathcal{W} - g(\mathcal{D}, P_\theta))$

## Varaits

- Joint : Combine shifts and sample into one vector and optimize for both together.
- Alternating : Alternate between optimizing with respect to sample and with respect to shifts.

# Implementation

## Language / Libraries / Architecture

- C/C++ using the PETSc/TAO framework
- Boost-geometry for the setup phase ($<$ 2% of runtime)
- FFTW for (MPI-rank local) fourier space convolutions
- 2-level MPI parallelism : MPI subcommunicators for concurrent instances of solver, each of which runs a TAO optimization problem with occasional syncing via PETSc VecScatters.
- Owing to the performance portable nature of PETSc, the non-error correcting version already runs on GPU's!

# Port Path & Goals

## GPU Port path :

- FFTW → cuFFT
- CUB for block-wise reductions
- remaining *for* loops → CUDA kernels

## Goals

- Focus on porting the TAO objective function & gradient routines in addition to some helper routines.
- Profile GPU versions to ensure that the port is efficient.
- If the single slice, single instance solver works, begin investigation of running concurrent solver instances.

# Change in strategy

## Surprises!

- Typically, $> 90\%$ of total time is spent in objective function/gradient routines, but on theta-gpu, $< 5\%$.
- Bottleneck : setting up non-contiguous data transfers between GPU arrays for bounds projection from estimated active set.

## Alternatives to explore

- Remove bound constraints $\rightarrow$ solution quality degrades.
- Re-evaluate for problem sizes of interest to APS beamlines ?
- If the active-set estimation/bound projection bottleneck still exists, explore alternate formulations like Augmented Lagrangian multiplier method.

# Results and Final Profile

| | 1 MPI Rank | | 4 MPI ranks | | 8 MPI Ranks | | 16 MPI Ranks | |
|---|---|---|---|---|---|---|---|---|
| | Total | % f/g | Total | % f/g | Total | % f/g | Total | % f/g |
| CPU-joint | 899.0 | 89.7 | 250.2 | 84.1 | 134.4 | 84.8 | 79.0 | 86.5 |
| GPU-joint | 128.8 | 2.9 | 66.2 | 5.8 | 37.8 | 12.4 | 58.9 | 21.3 |
| CPU-alt | 765.9 | 97.6 | 185.8 | 96.4 | 100.4 | 96.5 | 58.8 | 96.8 |
| GPU-alt | 22.5 | 12.7 | 11.8 | 17.4 | 9.5 | 30.0 | 22.4 | 54.1 |

Table: Analysis of total time as a function of MPI ranks for CPU and GPU solves and % of time spent in evaluation objective function and gradient.

# Problems encountered & Wishlist

## Problems encountered

- Some problems with using nvhpc which went away when using gcc and cuda with spack directly.

## Wishlist

- Better integration of nvhpc compiler within spack.

# Was it worth it ?

- Emphatic Yes! Presence of library developers and tool experts eased the path to porting and analyzing application. This, coupled with access to the latest generation GPU compute nodes helped in generating an informal performance model.
- We will follow up by taking a decision on the path forward for the application in terms of algorithm choice.
- Thank you to all the organizers for putting this event together and to both our mentors for all the advice and help in porting our application!

# References I

[Austin et al. 2019] Austin, Anthony P. ; Di, Zichao ; Leyffer, Sven ; Wild, Stefan M.: Simultaneous Sensing Error Recovery and Tomographic Inversion Using an Optimization-Based Approach. In: *SIAM Journal on Scientific Computing* 41 (2019), Nr. 3, S. B497–B521

[Jacobsen 2019] Jacobsen, Chris: *X-Ray Microscopy*. Cambridge University Press, 2019 (Advances in Microscopy and Microanalysis)