GPU-Based Acceleration for CT Image Reconstruction

Xiaodong Yu

Advisor: Wu-chun Feng

Collaborators: Guohua Cao, Hao Gong

VirginiaTech
Invent the Future

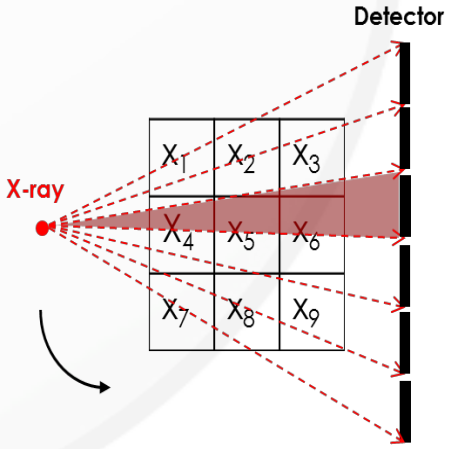
SyNeRG
synergy.cs.vt.edu

Outline

- Introduction and Motivation
- Background Knowledge
- Challenges and Proposed Solutions
- Experimental Results

Introduction and Motivation

- Computed Tomography (CT)
  - Indispensable imaging modality relying on multiple x-ray projections of the subject
  - Reconstruct 2D or 3D images through projection vector collected from detector
  - 60 million patients undergo a CT scan in 2002 in US
  - Disadvantage: carcinogenic nature of X-ray (should follow As Low As Reasonably Achievable principle)



Introduction and Motivation

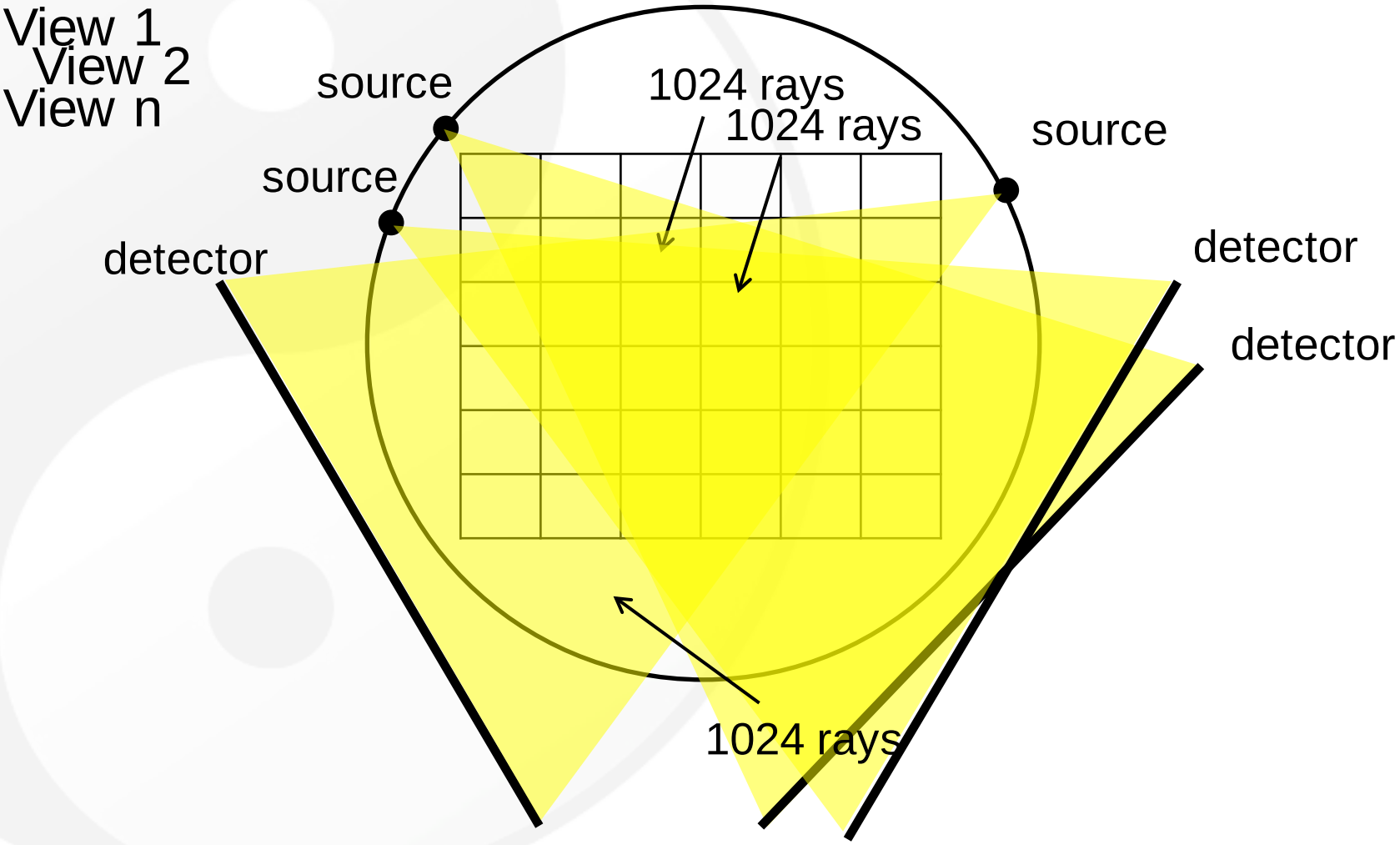
- CT image reconstruction approaches
  - Analytical methods
  - Iterative methods
- Filtered Backprojection (FBP) technique
  - Typical analytical algorithm, based on the Fourier Slice Theorem
  - Advantage: fast reconstruction
  - Disadvantage: requires many projection angles, hence more radiation dose for patients
- Algebraic Reconstruction Technique (ART)
  - Basic iterative algorithm
  - Advantage: produces better images with fewer projection
  - Disadvantage: high computational cost
  - One possible solution: map to HPC platform e.g. GPU

Background Knowledge

- Physical meaning of System Matrix
  - X-ray source and detector rotate around the object to get different views
  - The value of each system matrix entry is the weighting factor for corresponding pixel and corresponding ray
  - Column index of system matrix represent pixel index
  - Row index of system matrix represent ray index
  - Is a sparse matrix

Background Knowledge

- Physical meaning of System Matrix

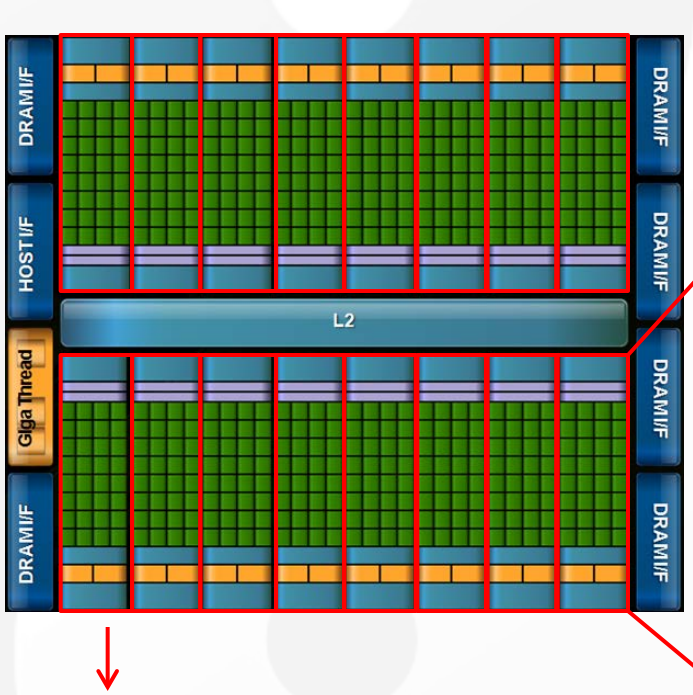


Background Knowledge

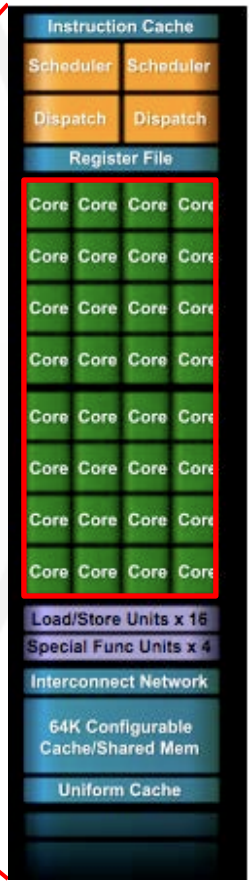
- Algebraic Reconstruction Technique (ART)
  - $f_j^{(n+1)} = f_j^{(n)} + \lambda_n \frac{p_i - S_i}{\sum_{j=1}^{N} w_{ij}^2} w_{ij}$
  - Where $S_i = \sum_{j=1}^{N} f_j^{(n)} w_{ij}$ and *n* correlated to *i*
  - *f* is the unknown image vector (N*1) of N=n*n pixels, *p* is the projection vector (M*1) of M rays, and *w* is the system matrix (M*N), whose element w_{ij} is the weight coefficient, representing the contribution of the *j*th pixel to the *i*th ray integral

Background Knowledge

- The ABC of GPUs
  - Two-level hardware parallelism



several independent STREAMING MULTIPROCESSORS (SMs)



32 SIMPLE PROCESSING CORES

SIMT processing: need for minimizing control flow divergence

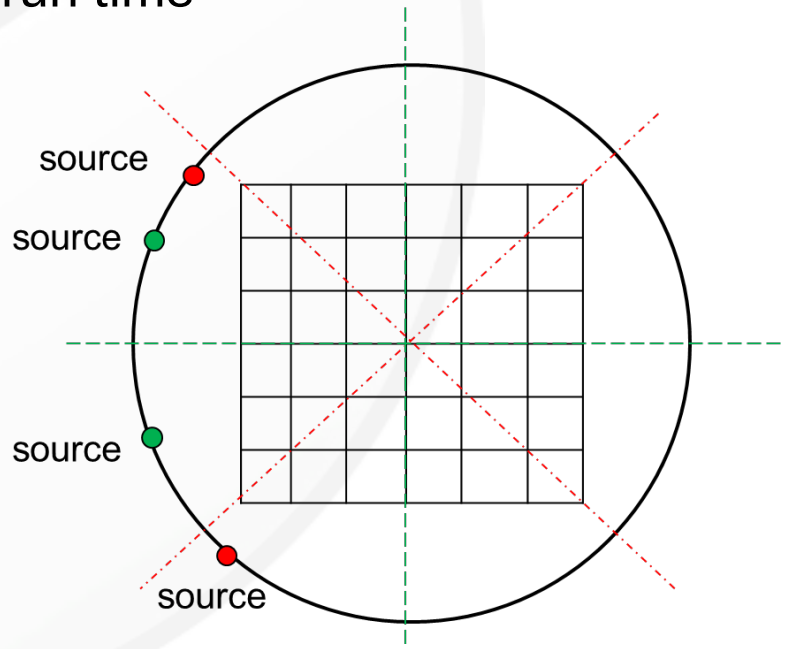
Challenges and Proposed Solutions

- Memory space requirement
  - A Tesla GPU usually has 6GB global memory
  - Uncompressed system matrix requires hundreds of GB memory space
  - Even a CSR-based compressed system matrix may require tens of GB memory space
  - Need to exploit further compression

View number	180		360	
Image size	Nonzero #	CSR size (GB)	Nonzero #	CSR size (GB)
512^2	157M	1.17	314M	2.34
1024^2	415M	3.09	830M	6.19
2048^2	1237M	9.22	2475M	18.44

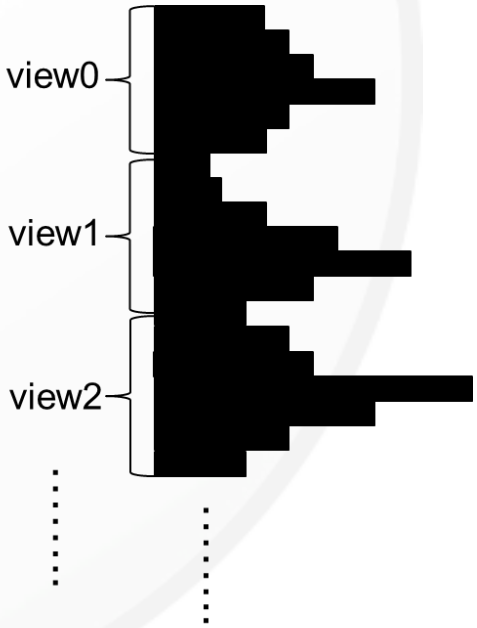
Challenges and Proposed Solutions

- Symmetry-Based Compression
  - Two kind of view symmetry: rotational and reflection
  - Only weighing factors whose corresponding views are within one 45 degree need to be recorded, accordingly, only one eighth of system matrix need to be transfer to GPU, all others can be calculated in the run time



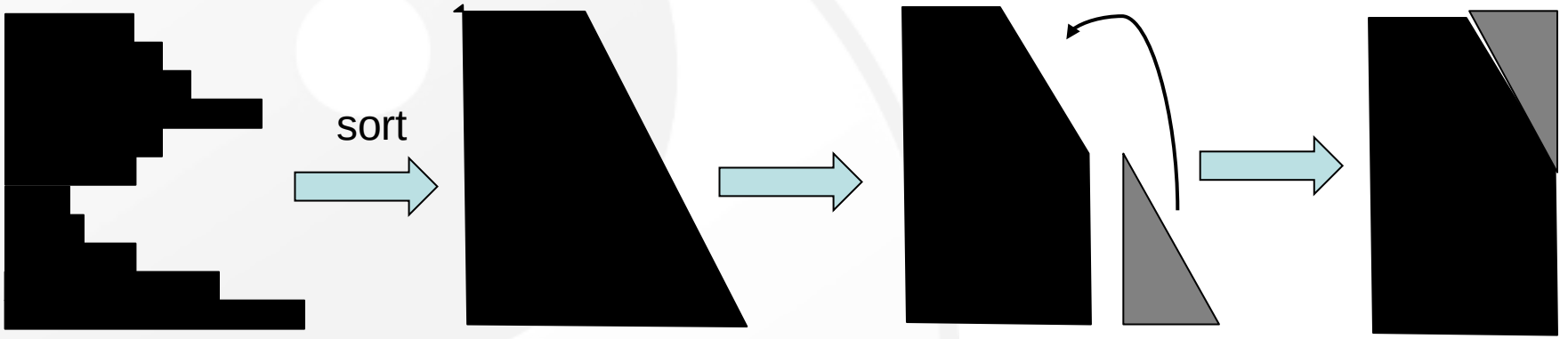
Challenges and Proposed Solutions

- Load Imbalance
  - Each row of system matrix is assigned to one thread
  - Different rows have different numbers of non-zero element, lead to different threads have different workloads
  - Load imbalance degrade the performance

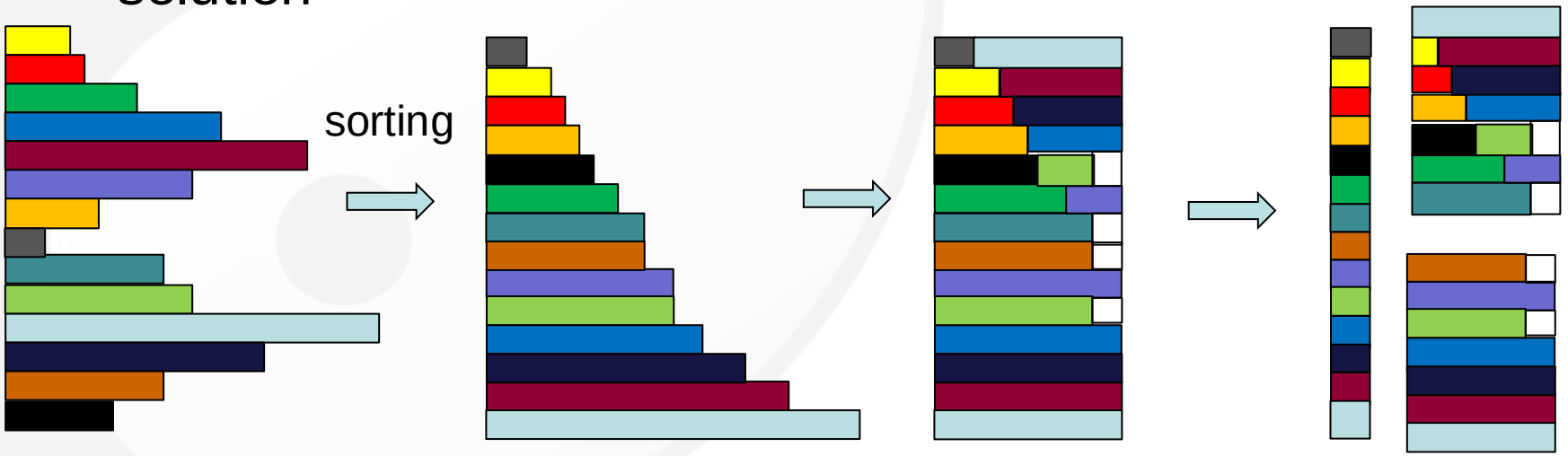


Challenges and Proposed Solutions

- Intuitive idea



- solution



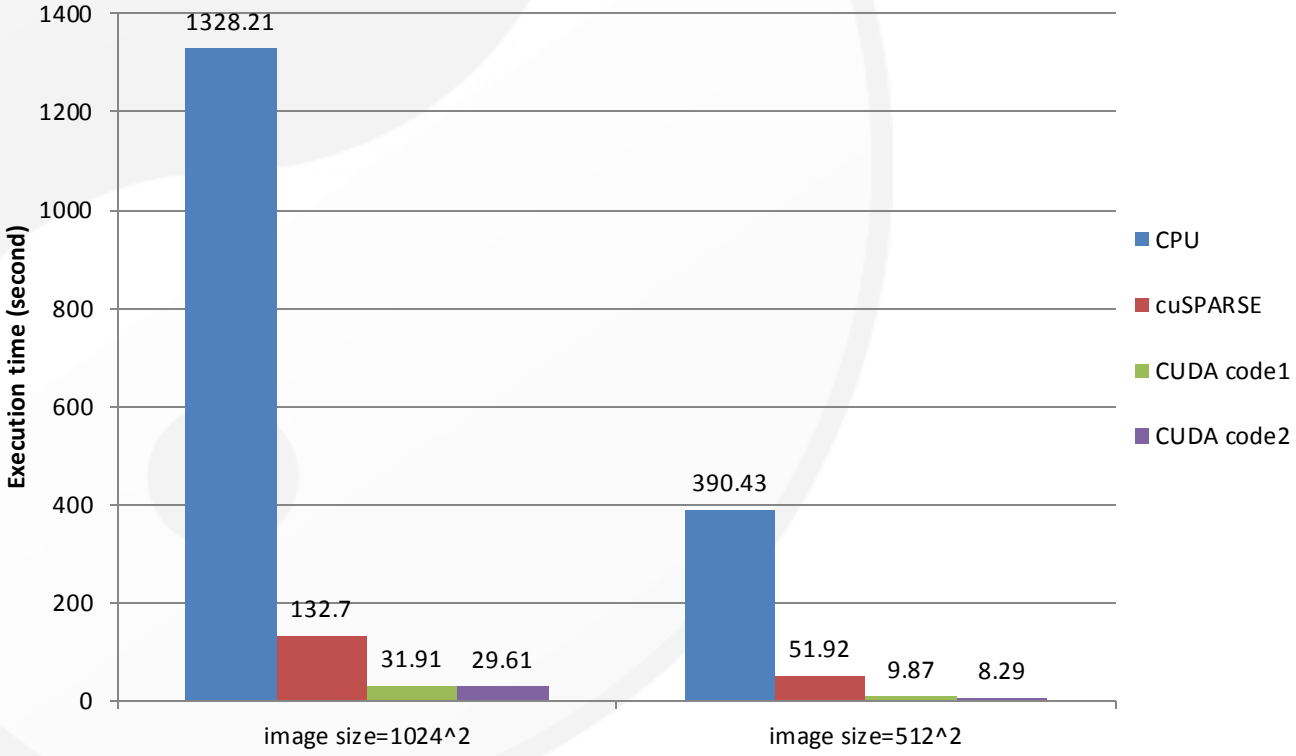
Experimental Results

- Symmetry-Based Compression

View number	360		720	
Image size	CSR size (GB)	w/ Sym-based compression(GB)	CSR size (GB)	w/ Sym-based compression(GB)
256^2	1.01	0.12	2.03	0.25
512^2	2.34	0.29	4.82	0.60
1024^2	6.19	0.77	11.97	1.50
2048^2	18.44	2.31	37.41	4.68

Experimental Results

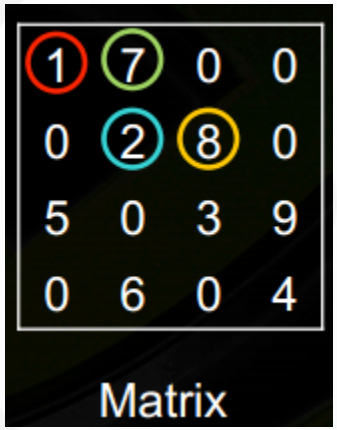
- Performance Comparisons
  - Two image sizes: 1024*1024, 512*512
  - Both have 720 views
  - Both do 10 iterations

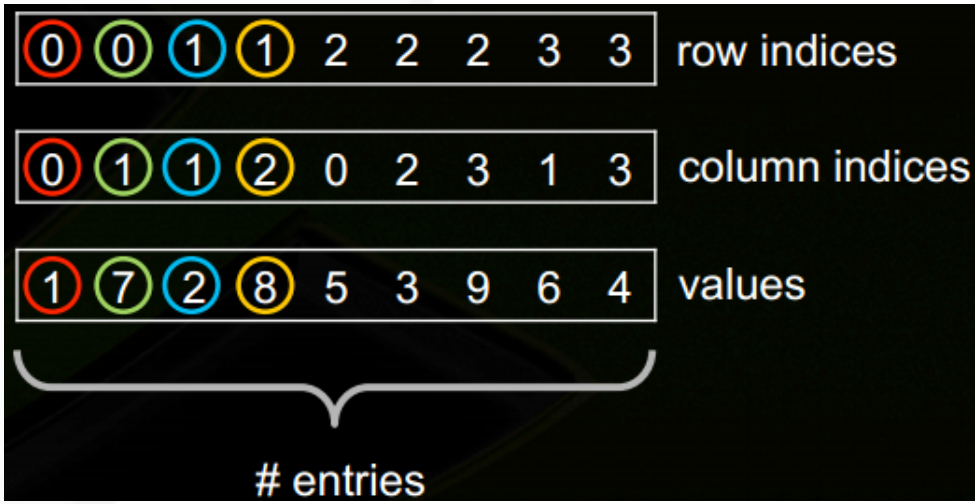


Back up

COO

- Coordinate list (COO)
  - Basic format for sparse matrix compression
  - stores a list of (***row, column, value***) tuples
  - ***row, column, value*** are three arrays
    - ***row*** stores the row indices of non-zero entries
    - ***Column*** stores the column indices of non-zero entries
    - ***Value*** stores the values of non-zero entries





CSR

- Compressed Sparse Row (CSR)
  - Another basic format for sparse matrix compression
  - Avoid the off-chip memory pressure of COO (accessing row indices)

- Consists of three arrays: ***val, col_ind, row_ptr***
  - ***val*** stores the values of non-zero entries
  - ***col_ind*** stores the column indices of non-zero entries
  - ***row_ptr*** stores the start pointers of the nonzeros of the rows