



Parallel Multifrontal  
Sparse LU  
Factorization Based on  
UMFPACK

*Mohsen Aznaveh*

*Advisor: Dr. Tim Davis*

*SPARSE DAYS 2020 online*

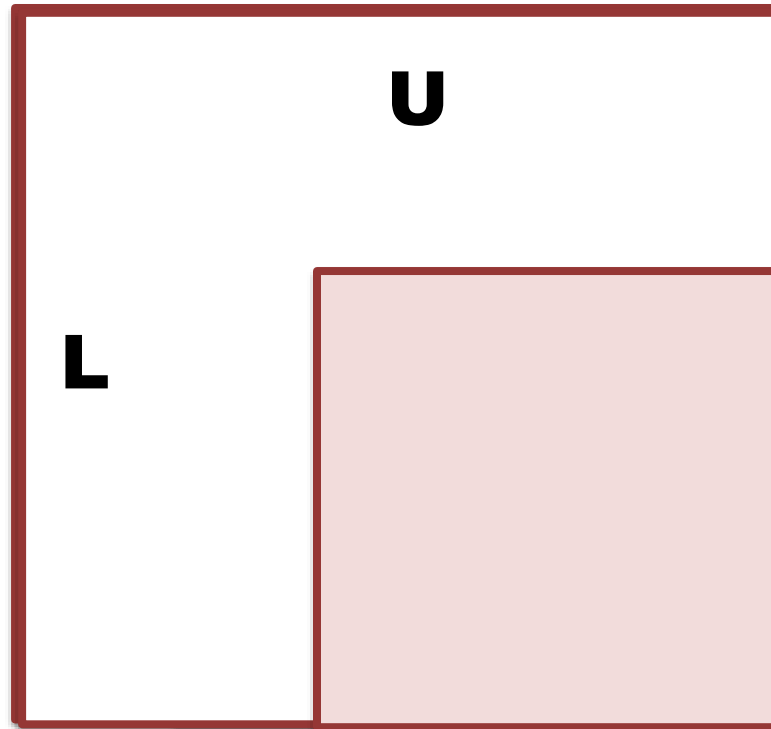
*Nov 2020*

# Introduction

- This is a work in progress
    - for five years so far
  - The ideas come from UMFPACK
    - Also call UMFPACK for symbolic analysis
    - Although developed a code from scratch
  - The goal was to have a code suitable for parallelization
-

# LU factorization

Right looking method



# UMFPACK

- Unsymmetric Sparse LU factorization
    - Unsymmetric pattern
  - Multifrontal method
  - Rectangular fronts
  - Sparsity is also important in pivot choice
  - Intrinsically sequential
  - Good performance in practice
-



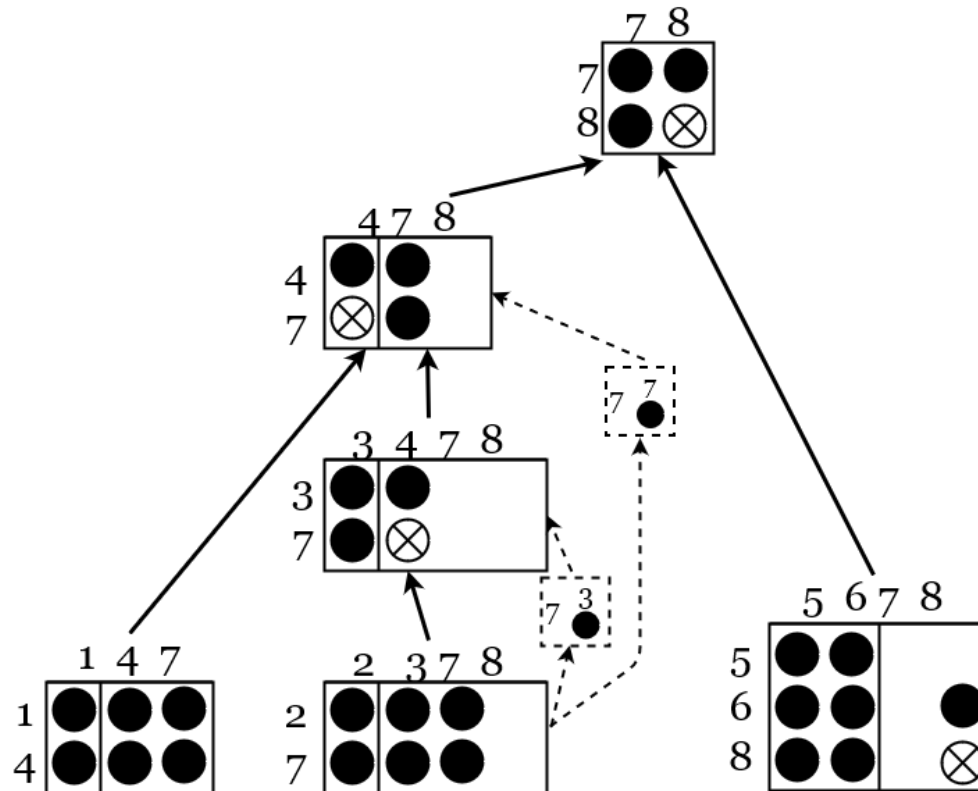
# Ongoing development

# Symbolic analysis

- Forming a suitable elimination tree
  - Permuting the matrix
  - Postordering the tree
  - Augmented tree
  - Computing the staircase structure
  - Relaxed amalgamation

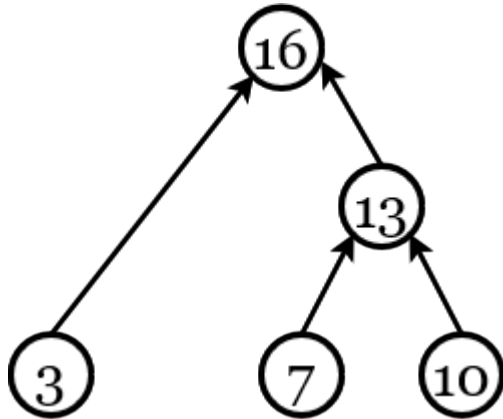
**Done!**  
**Not Parallel**

# Unsymmetric Assembly tree

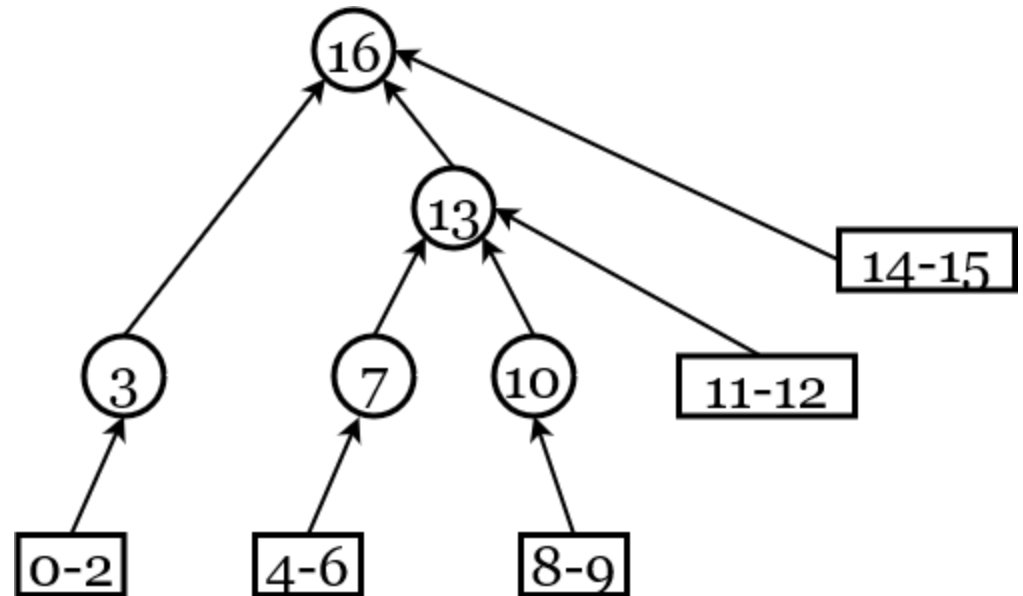


# Augmented Tree

original tree



postordered

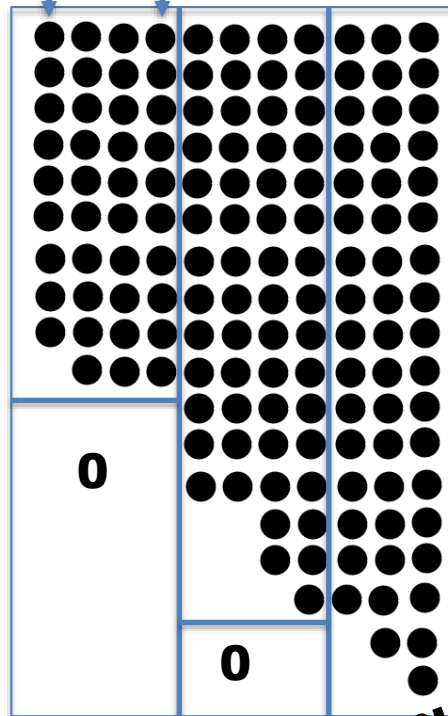




# Front computation

Pivotal columns are given by the symbolic analysis

Pivotal rows are a set union



After each panel

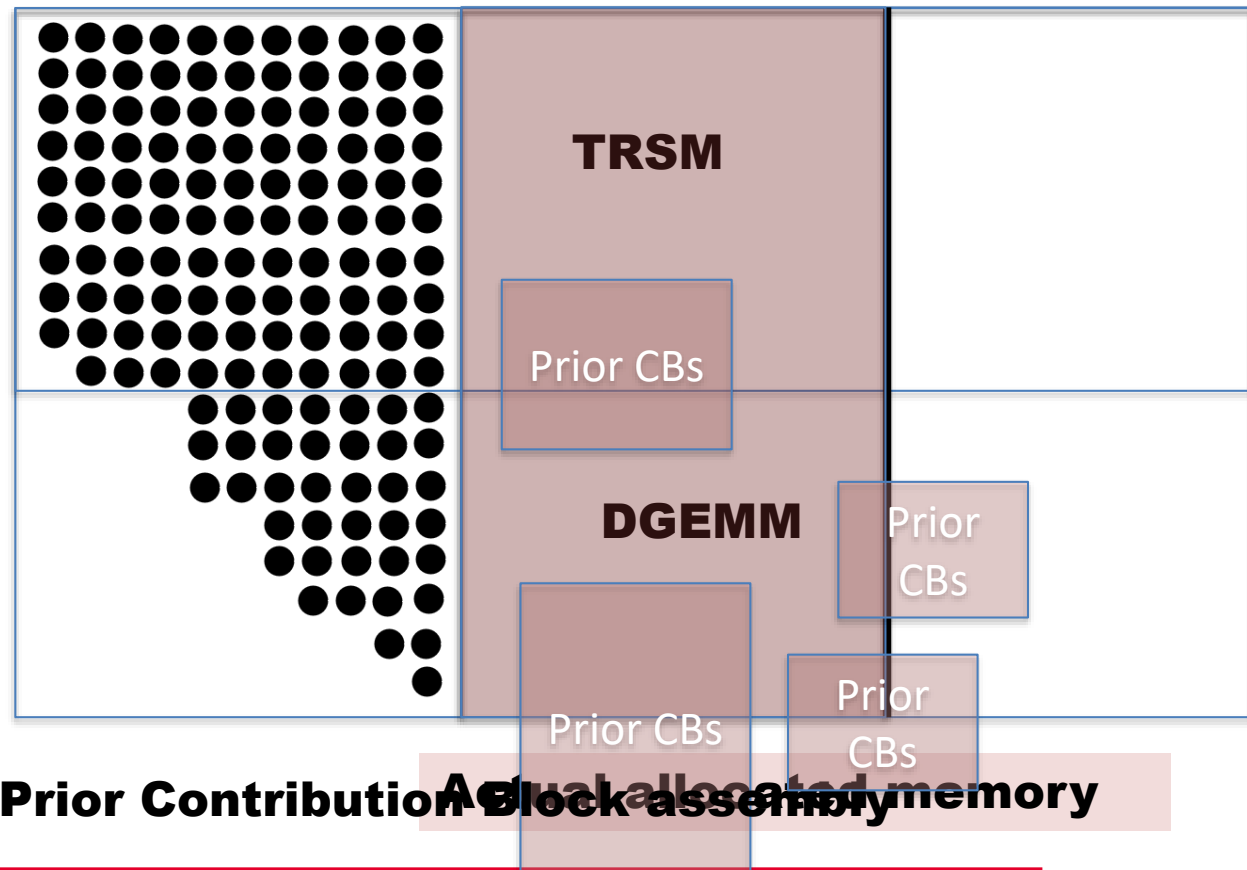
- TRSM and DGEMM
- Firstly, the pivot is updated on value and sparsity

DGER for the rest of the panel

We have an upper bound on number of CB columns

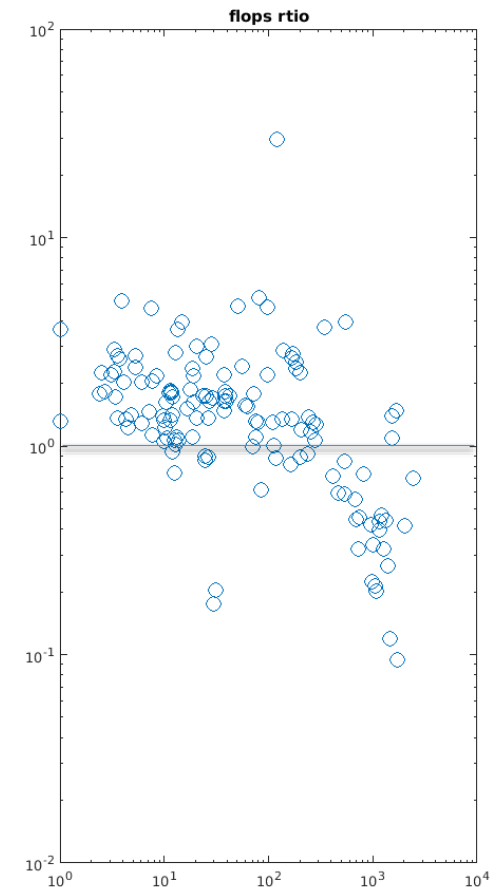
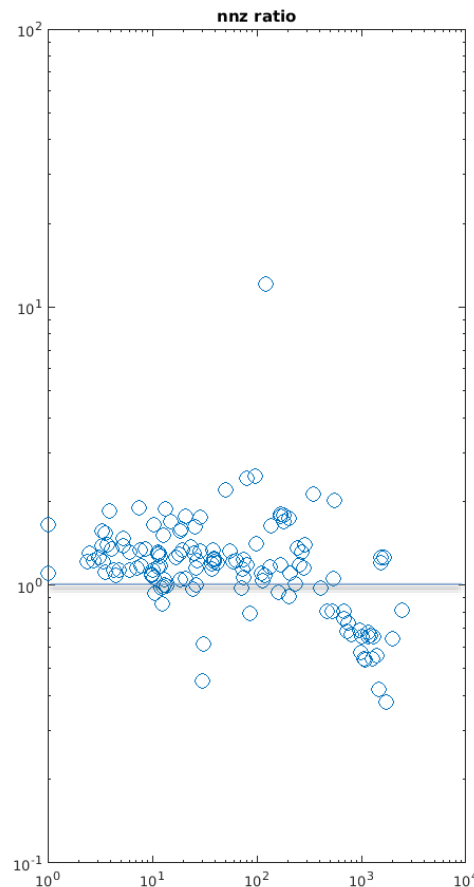
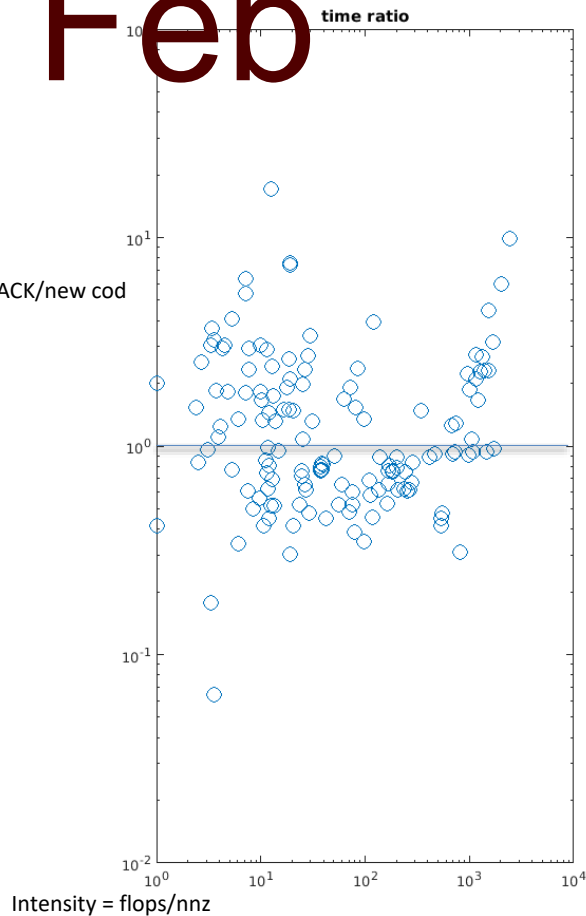
No column permutation

# Front computation



# Some Results from Feb

UMFPACK/new cod



# Parallelization

- Fronts can be parallel based on the etree
  - Using dense BLAS kernels
  - Different stages
-

# Challenges in Parallelization

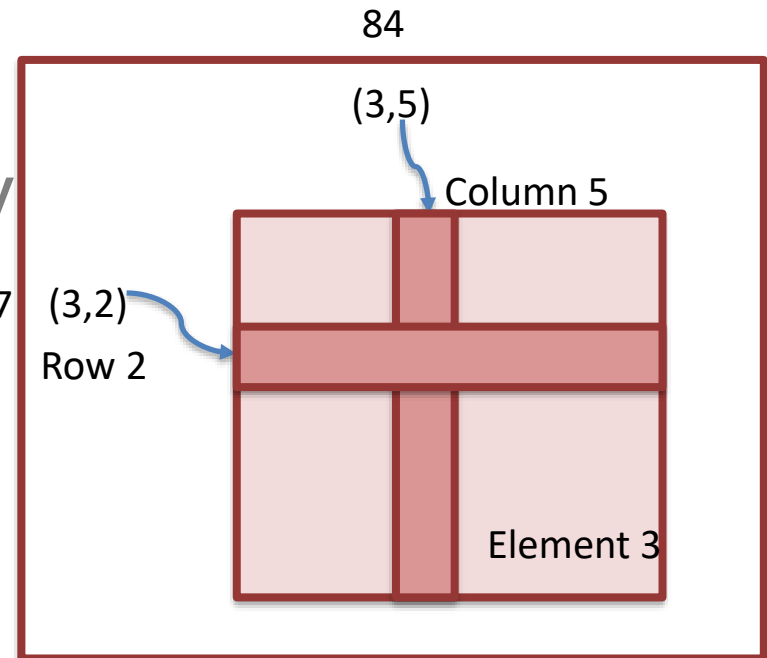
- Like UMFPACK we use Tuples for Rows and Columns  
(CB number, Row/Col number)
  - Rows are independent
  - Column tuples are a challenge
- Load balancing

Fixed

---

# Tuples

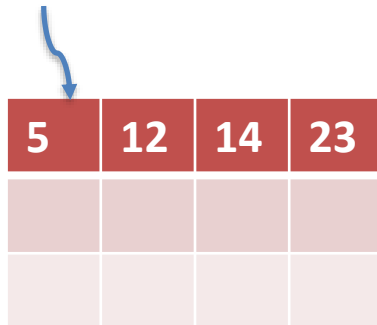
- For each row and col we have a list
- Typically short
- Is used for assembly
- And degree update <sup>37</sup>



# Our solution for col tuples

- Sort columns of elements in the time of creation
- Save the least active column

Least active column

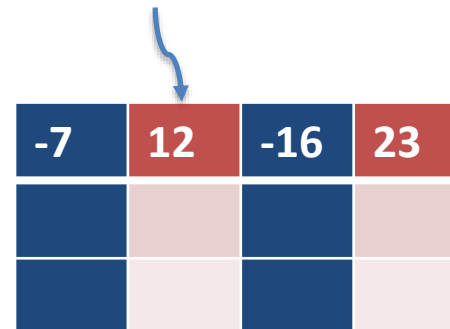


5	12	14	23

After assembling  
cols 5,14



Least active column



-7	12	-16	23

# Our solution for col tuples...

- A list of active descendent of each front on a min-heap
    - The key of the heap is the least active column
  - Pivotal column assembly would be extracting from the heap
-

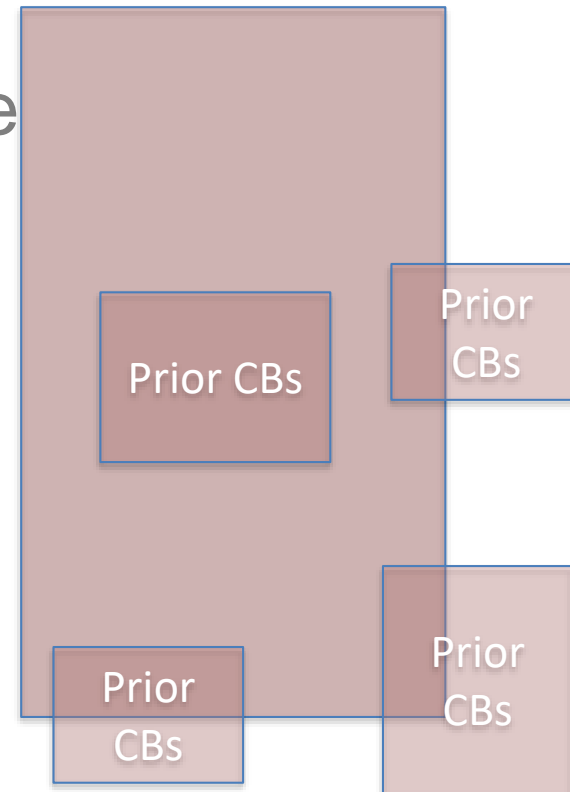


# Our solution for col tuples...

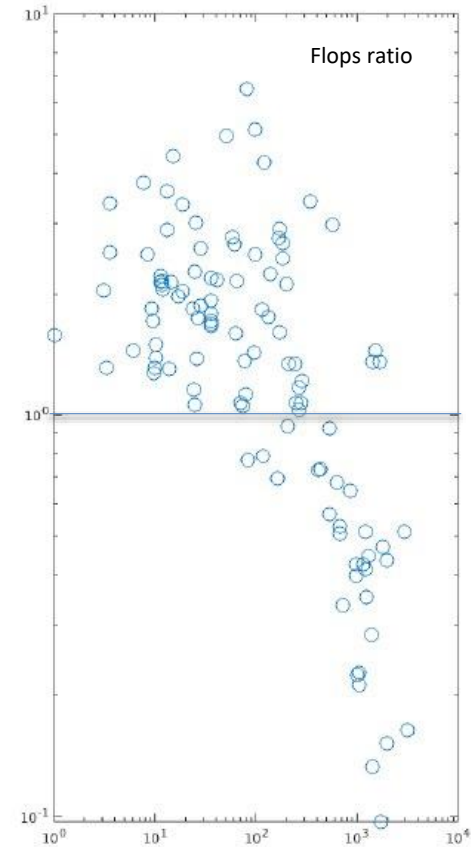
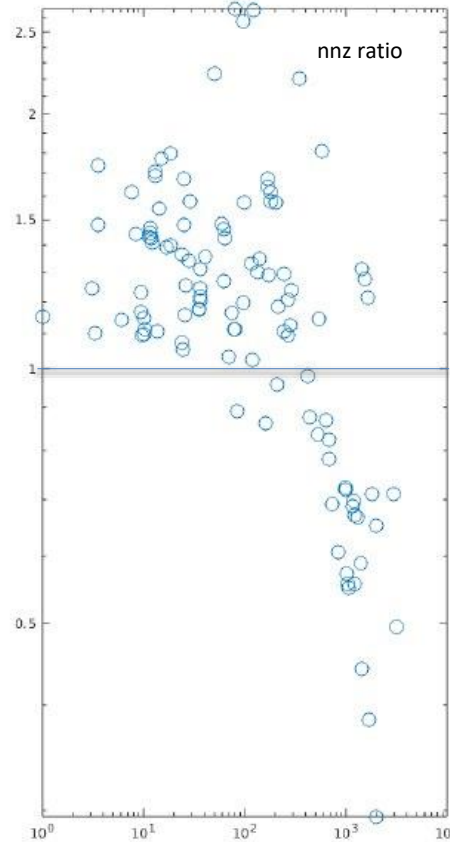
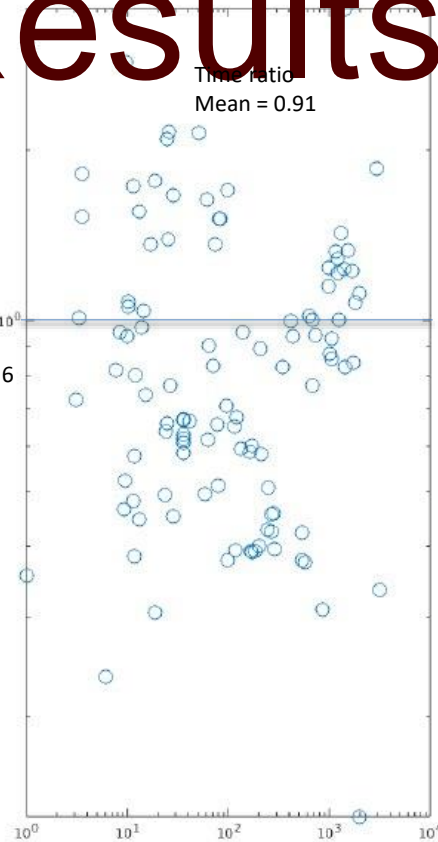
- After making the current CB we make a hash of column
    - Local for the front so no dependency
    - Finding a column in prior CB
      - Look the hash
      - Binary search
-

# Our solution for col tuples...

- We use Hash
  - Row degree update
  - Prior CB assembly



# New Sequential Results



UMFPACK/new code  
nnz > 10000  
Pattern symmetry < .6

Intensity = flops/nnz

A faint, dark red background image of a large building dome, likely a state capitol building, centered behind the text. The dome is the central focus of the background, with its base and surrounding architectural details visible but less distinct due to the low opacity.

**Thanks!**

**Any questions?**