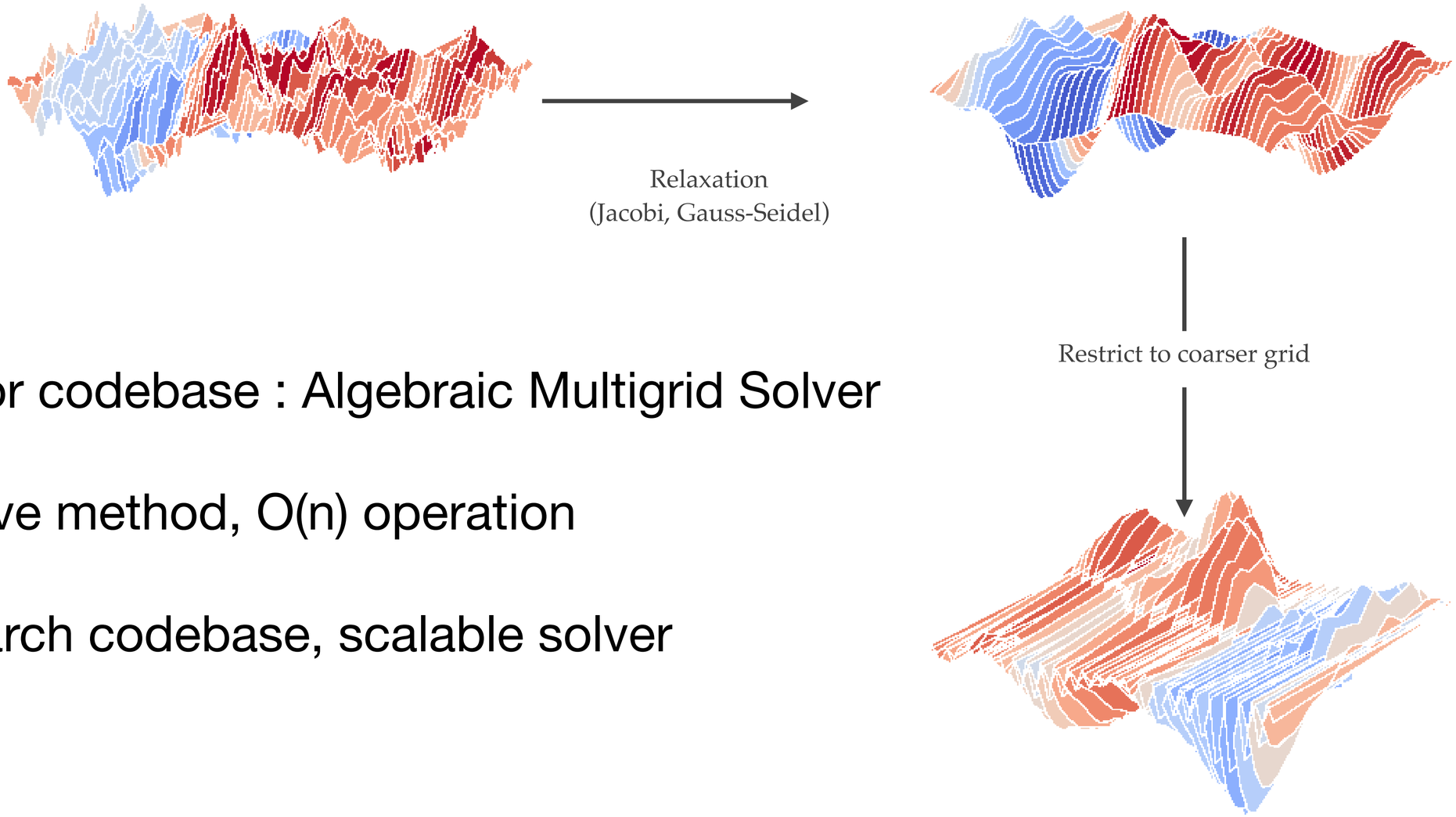
Improving I/O for Sparse Linear System Solver

12/04/2020

Amanda Bienz

Sparse Solver

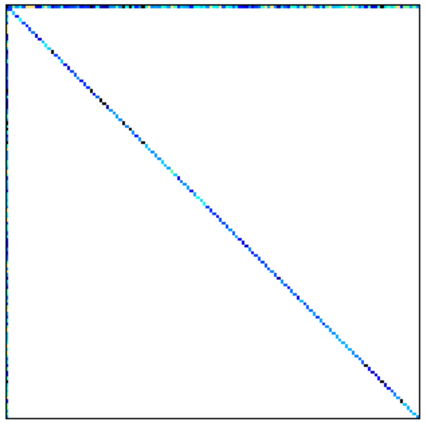


- RAPtor codebase : Algebraic Multigrid Solver
- Iterative method, O(n) operation
- Research codebase, scalable solver

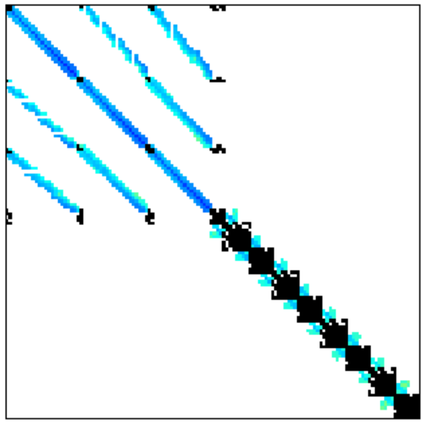
RAPtor: parallel algebraic multigrid. Bienz, Amanda and Olson, Luke N. 2017. Release 0.1. https://github.com/raptor-library/raptor

Sparse Matrices

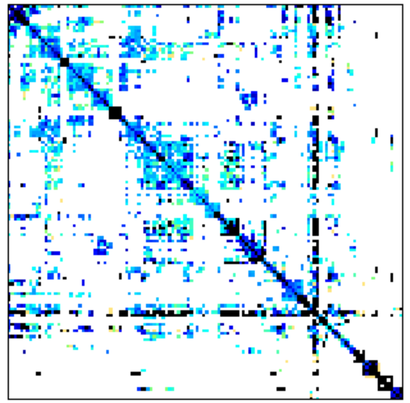
- Solver research : application agnostic
- Sparse systems : performance is dependent on sparsity pattern
- Need to be able to test large variety of systems to get full picture of performance
- Suitesparse matrices : must be read in with parallel I/O



bundle_adj



consph



bmw7st_1

Images from : https://sparse.tamu.edu

RAPtor Standard I/O

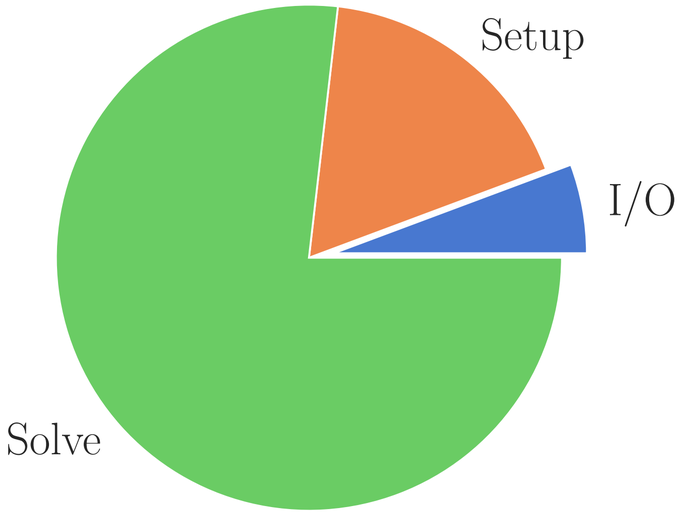
- Reading in a binary matrix (converted from matrix market to PetscBinary format)

Algorithm 1: Original Parallel I/O Algorithm

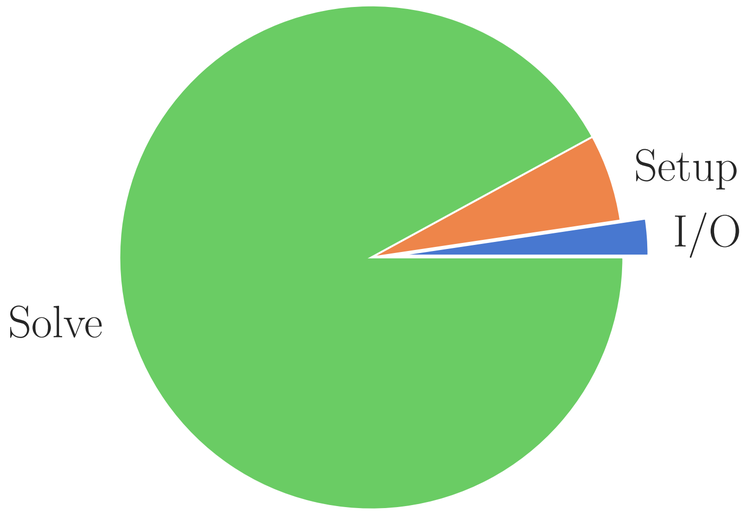
```
Read header
Seek to first row
for i ← 0 to n do
    Read row size
Seek to first colidx
for i ← 0 to nnz do
    Read column index
Seek to first data val
for i ← 0 to nnz do
    Read data value
```

RAPtor Standard Performance

- What percent of program cost is due to I/O (vs AMG hierarchy setup and iterative solve)?



consph



bmw7st_1

128 Processes

Parallel I/O Optimizations

- Best performance : small number of large I/O accesses

Algorithm 1: Original Parallel I/O Algorithm

```
Read header
Seek to first row
for i ← 0 to n do
    Read row size
Seek to first colidx
for i ← 0 to nnz do
    Read column index
Seek to first data val
for i ← 0 to nnz do
    Read data value
```

Parallel I/O Optimizations

- Replace loop of small reads with larger contiguous read

Algorithm 1: Optimized Parallel I/O Algorithm

Read header

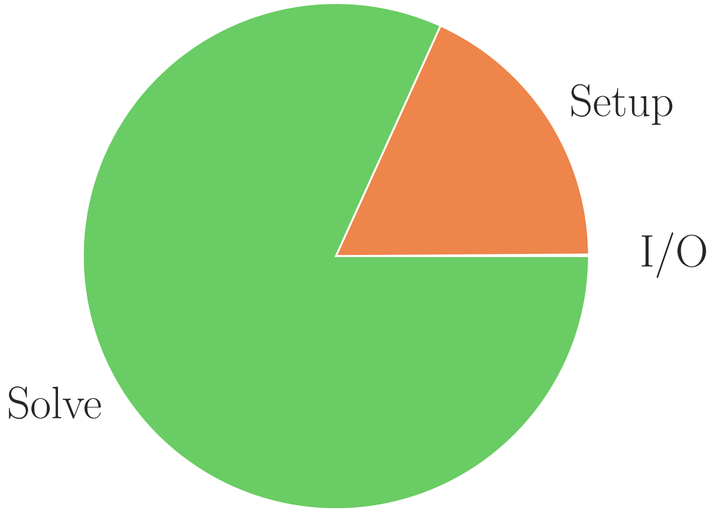
Seek to first row
Read n row sizes

Seek to first colidx
Read **nnz** column indices

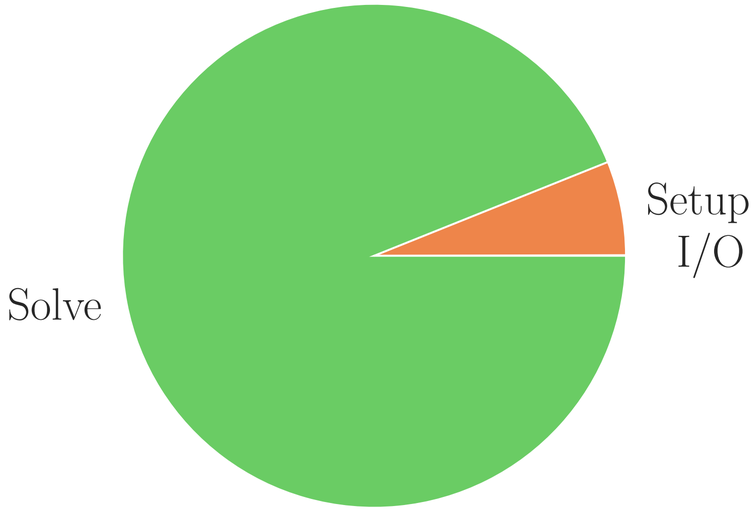
Seek to first data val
Read **nnz** data values

RAPtor Optimized Performance

- What percent of program cost is due to I/O (vs AMG hierarchy setup and iterative solve)?



consph



bmw7st_1

128 Processes

MPI I/O

Algorithm 1: Optimized Parallel I/O Algorithm

Read header

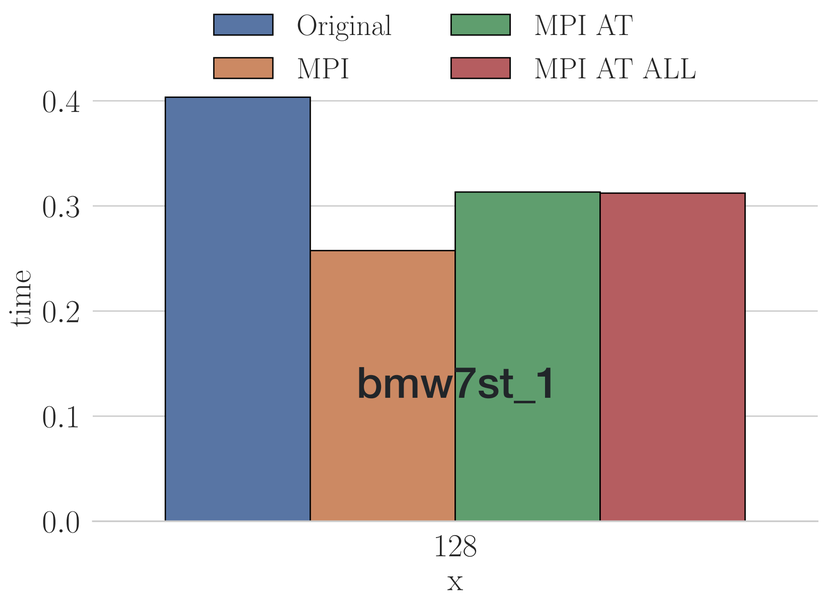
Seek to first row
Read n row sizes

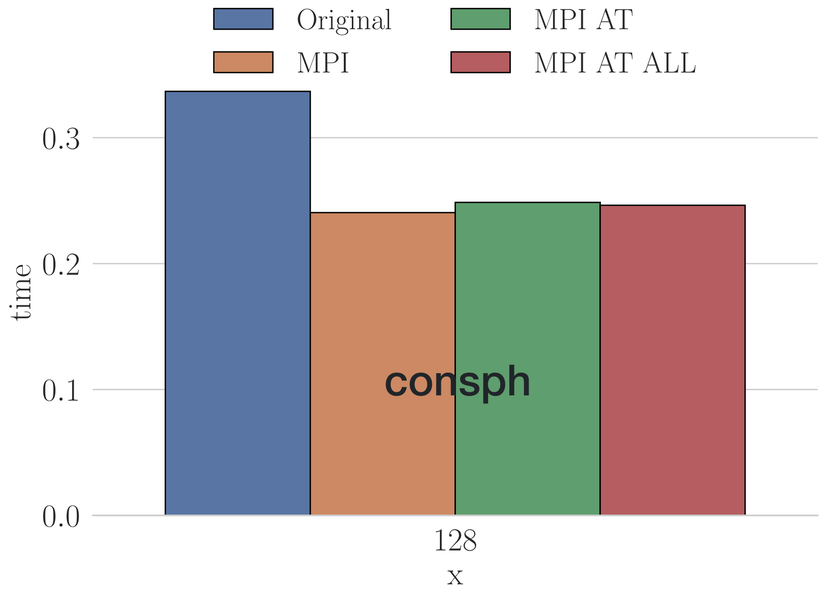
Seek to first colidx
Read `nnz` column indices

Seek to first data val
Read `nnz` data values

Different Tests :

1. **MPI_File_seek(...)**
   MPI_File_read(...)
2. **MPI_File_read_at(...)**
3. **MPI_File_read_at_all(...)**





Future Directions

- Test larger matrices (more non-zeros per process) to get accurate analysis of where majority of time is spent
- Test different partitions (not contiguous rows)
- Improve I/O performance for writing sparse matrices to file (useful for checkpointing)

Presentation Advice :

- Tell a story : motivate project, describe what you did, and talk about how this could extend in the future
- You don’t have to describe everything you have done (only 5 minute presentation)… pick out what you think is the most interesting and talk about this
  - Note : I talked very little about MPI I/O (but it is a large part of my report)

Presentation Advice :

- Don’t put too much text on a slide
- A picture is worth 1,000 words
- Try to think of a way to add pictures that describe your work
  - If you can’t think of a way, send me a note and I will help brainstorm

Presentation Advice :

- Limit code on slides
- If you are describing an algorithm, try to do with either with a picture or an algorithm environment, rather than code
- If you feel that code is necessary to describe your work, limit to as little code as possible
- Again, I’m happy to help you come up with ways to present your work without a bunch of code

Presentation Advice :

- Try not to read the slides exactly
  - Easiest if you have pictures you are describing
- Explain any figures you have in detail
  - You have worked with these a lot and understand everything about them, but we are seeing them for the first time
- You can tell the class what to take away from a slide
  - Highlight important parts of slide