Recent Advancements in SZL for I/O Efficient Analysis and Visualization of Overset Grid Results

A presentation for the 12th Symposium on
Overset Composite Grids and Solution Quality

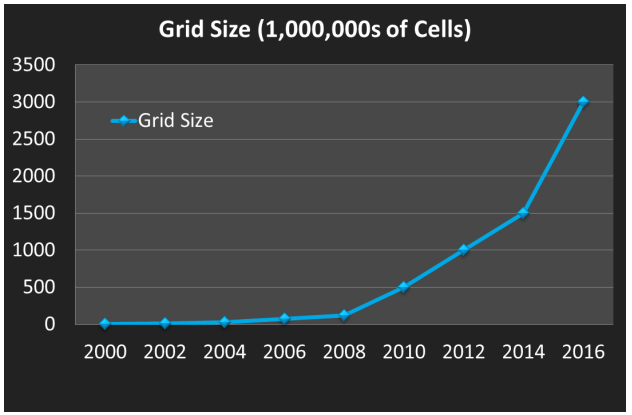
Scott T. Imlay, Durrell K. Rittenberg, and
Craig A. Mackey

tecplot® Master the View

Outline

- Trends (hardware and data)
  - Data Growth (Grand Challenge)
  - Hardware trends
- SZL
  - General Algorithm
  - Progress since 2012
- Results

- Goal: Improve performance and reduce memory requirements for large and/or remote data.

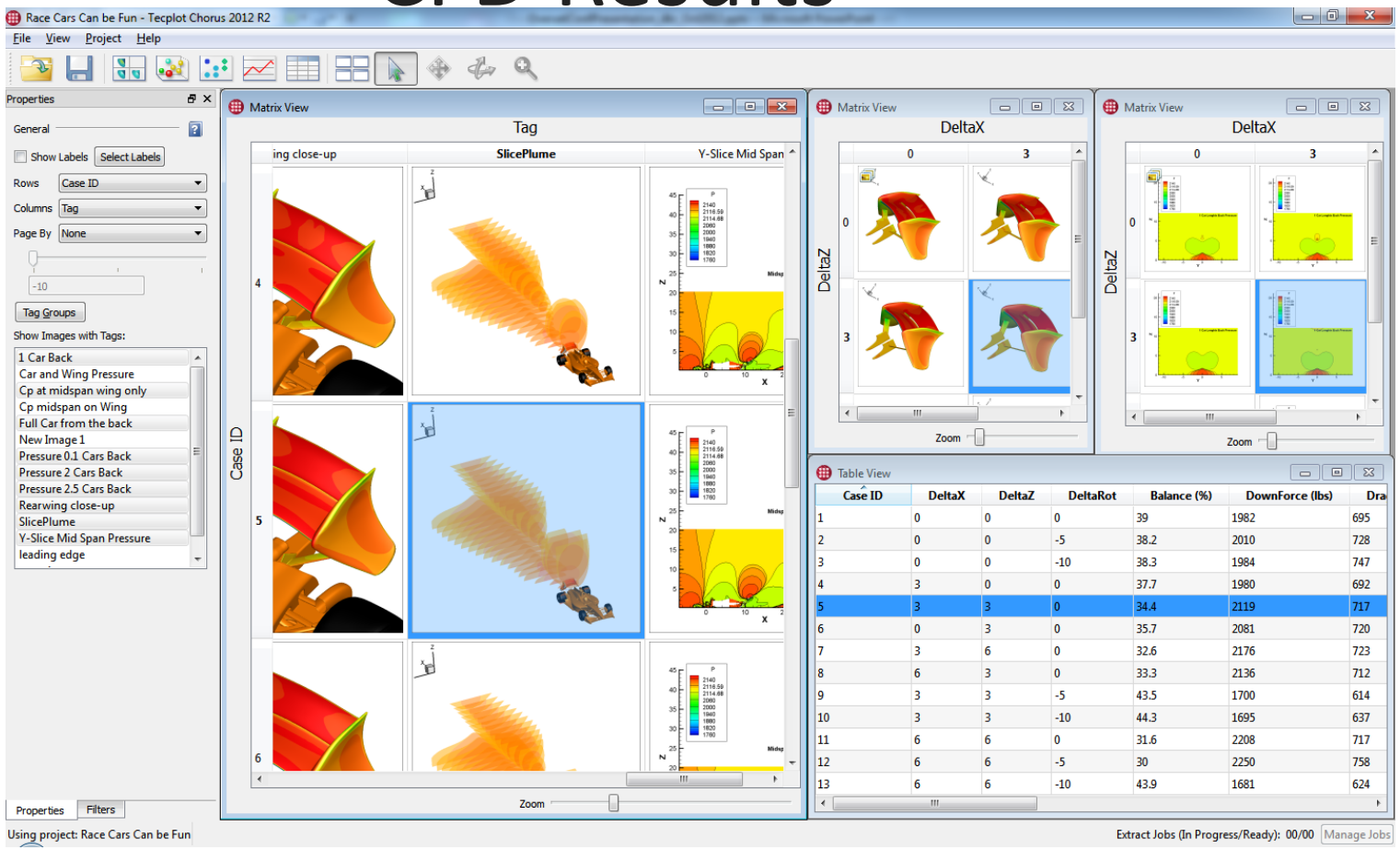


tecplot® Master the View

CFD Dataset Size Growing with Moore's Law

- Wide range in length scales
- Resolution of grid (# of grid points) constrained by computer performance (growing with Moore's law)

Tecplot Chorus For Collections of CFD Results



Evaluating overall system performance and allowing engineers to compare results quickly

tecplot® Master the View

NASA CFD 2030 Roadmap

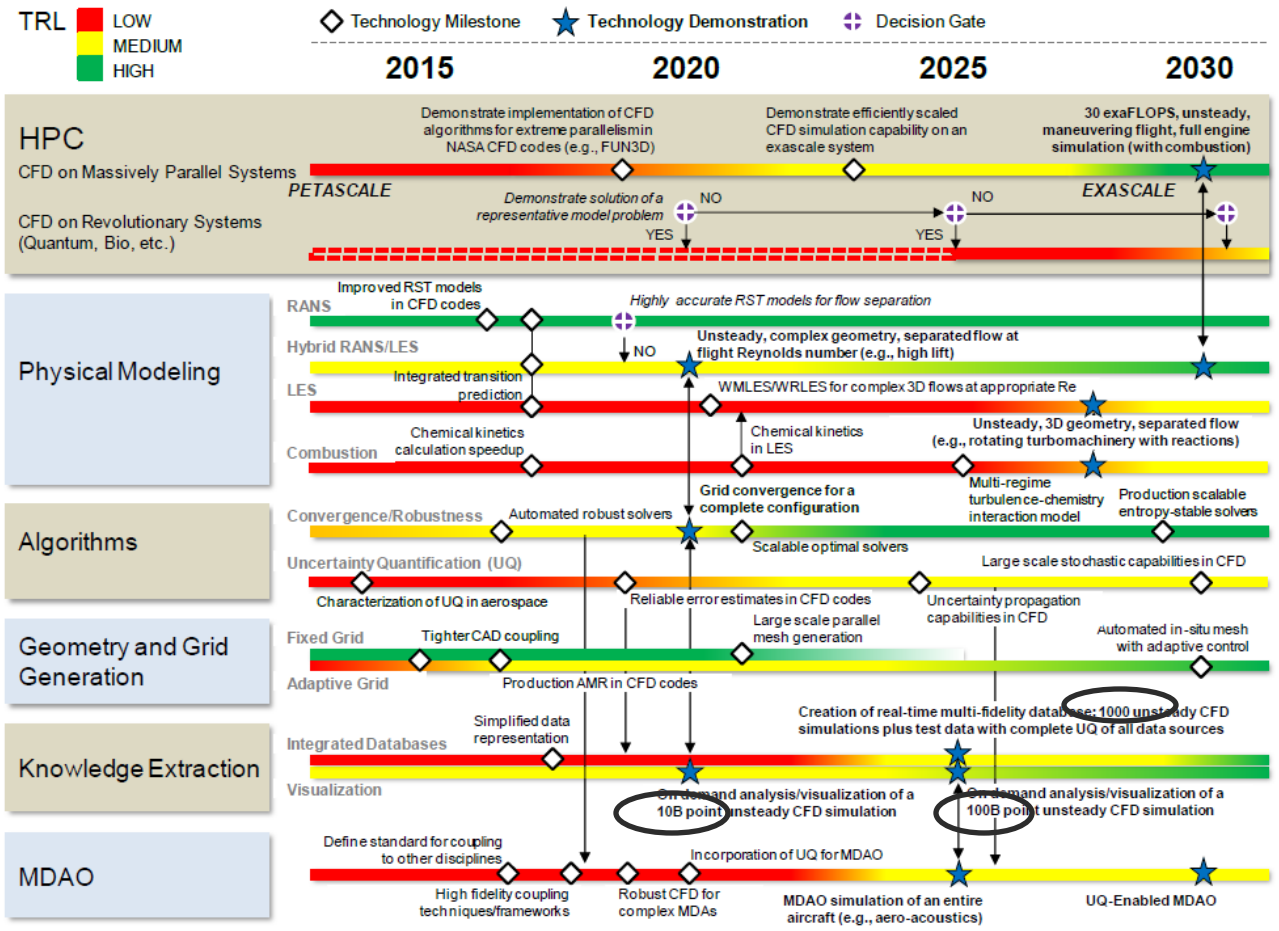


Figure 1. Technology Development Roadmap

tecplot® Master the View

Tecplot Grand Challenge: Visualize a Trillion Cell FE Solution

On This Computer

< $10,000

2 Intel Xeon (8 cores each)
128GB memory
16TB Raid5 storage

NOT this computer

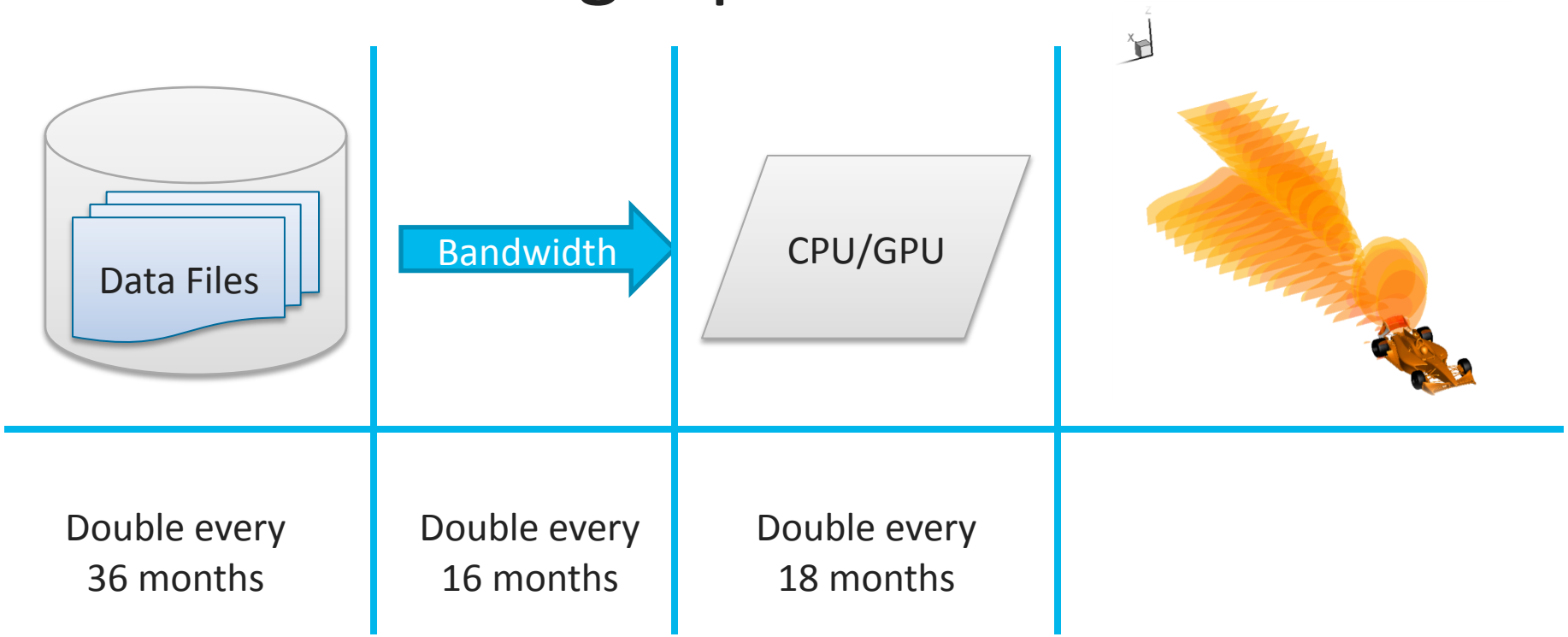
> $10M ?

tecplot® Master the View

Why a Trillion Cells?

- Yes, we know no one is doing this yet
  - Will be by 2030, if growth continues with Moore's law
  - We wish to insure that our software is written to handle future growth
    - For example: algorithm scaling is critical – nothing can scale O(N)

Data Processing Pipeline

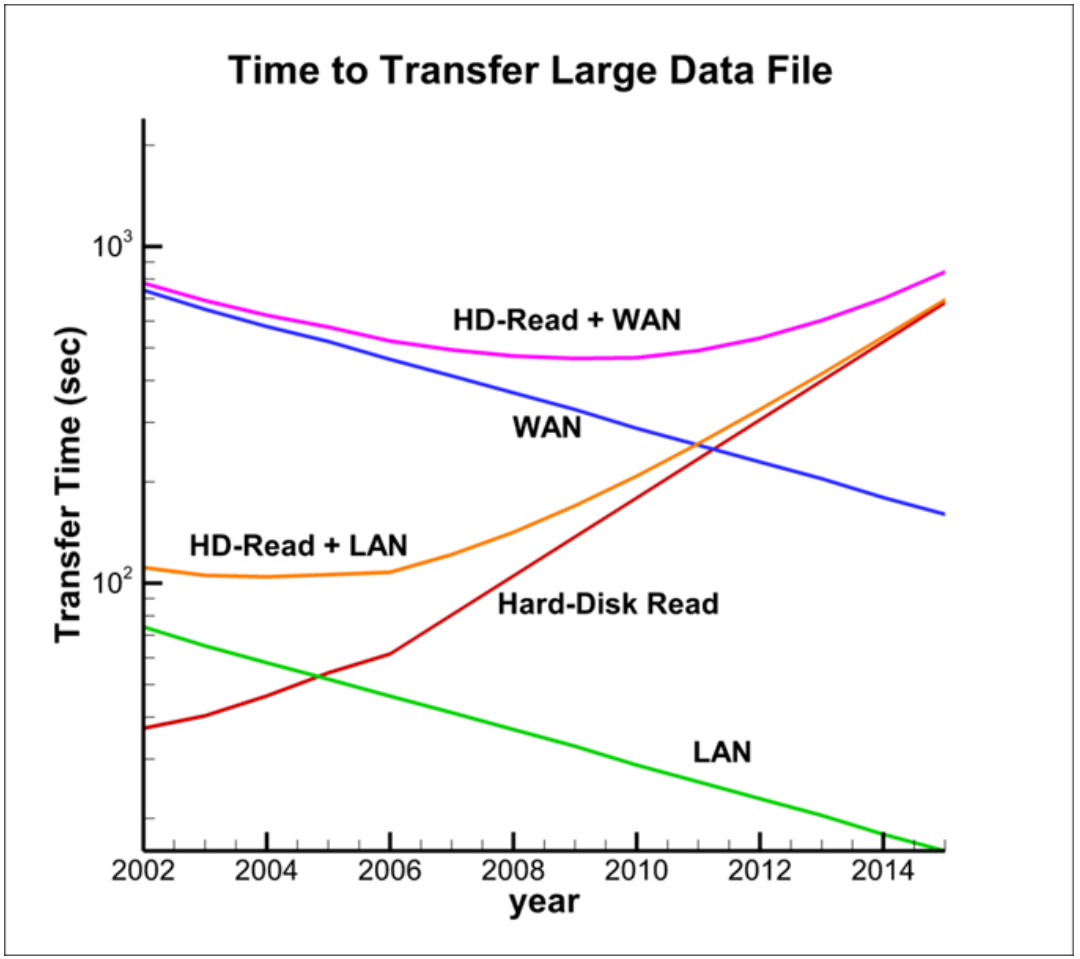


Data IO is the current rate determining step in the visualization pipeline.

tecplot® Master the View

Consequence of Data I/O Bottleneck

- Disk read performance growing slower than grid size
- Current visualization architectures will perform worse as time goes on!

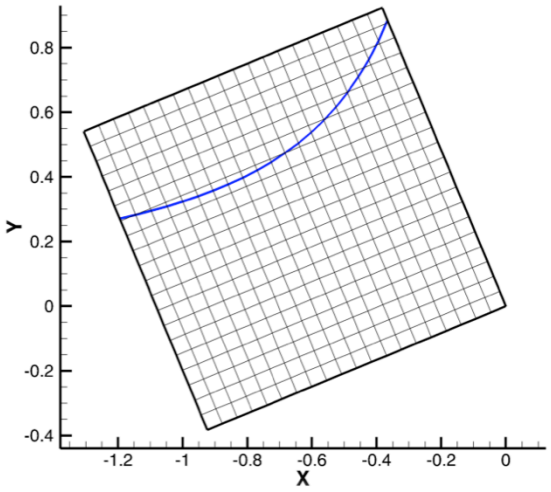


tecplot Master the View

Overcoming Data Transfer Bottleneck
Our Solution

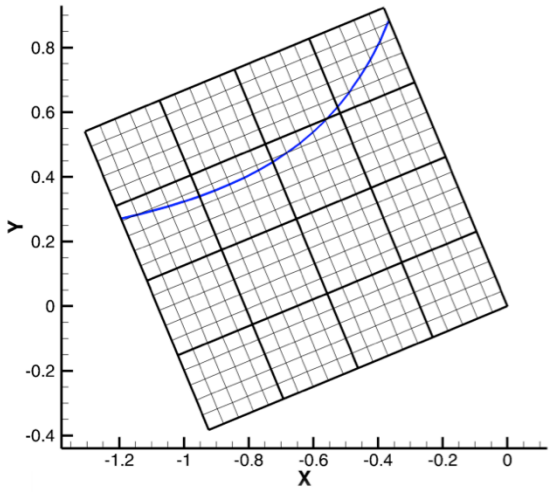
- Reduce the amount of data you read!
  - Must scale sub-linearly with the size of the grid
- Subzone Load-on-Demand (SZL)
  - Save indexed volume data file
  - Load only the data you need (Lazy Loading)

How Does SZL Work?



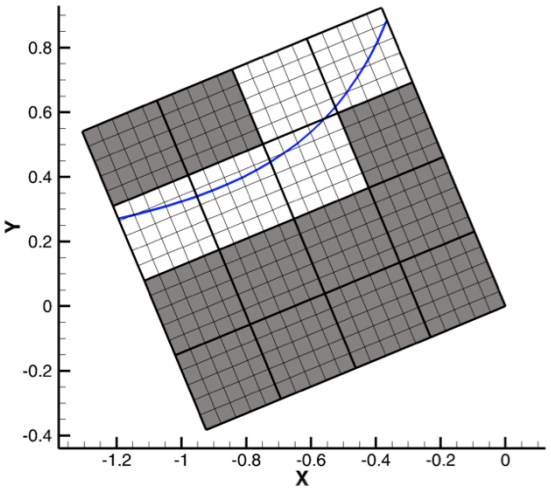
Example 2D Contour Line

- Current Methodologies require loading data for zone
- For Large data loading can be time intensive



Domain can be indexed

- Decomposition of domain into smaller subdomains
- These subdomains can be indexed

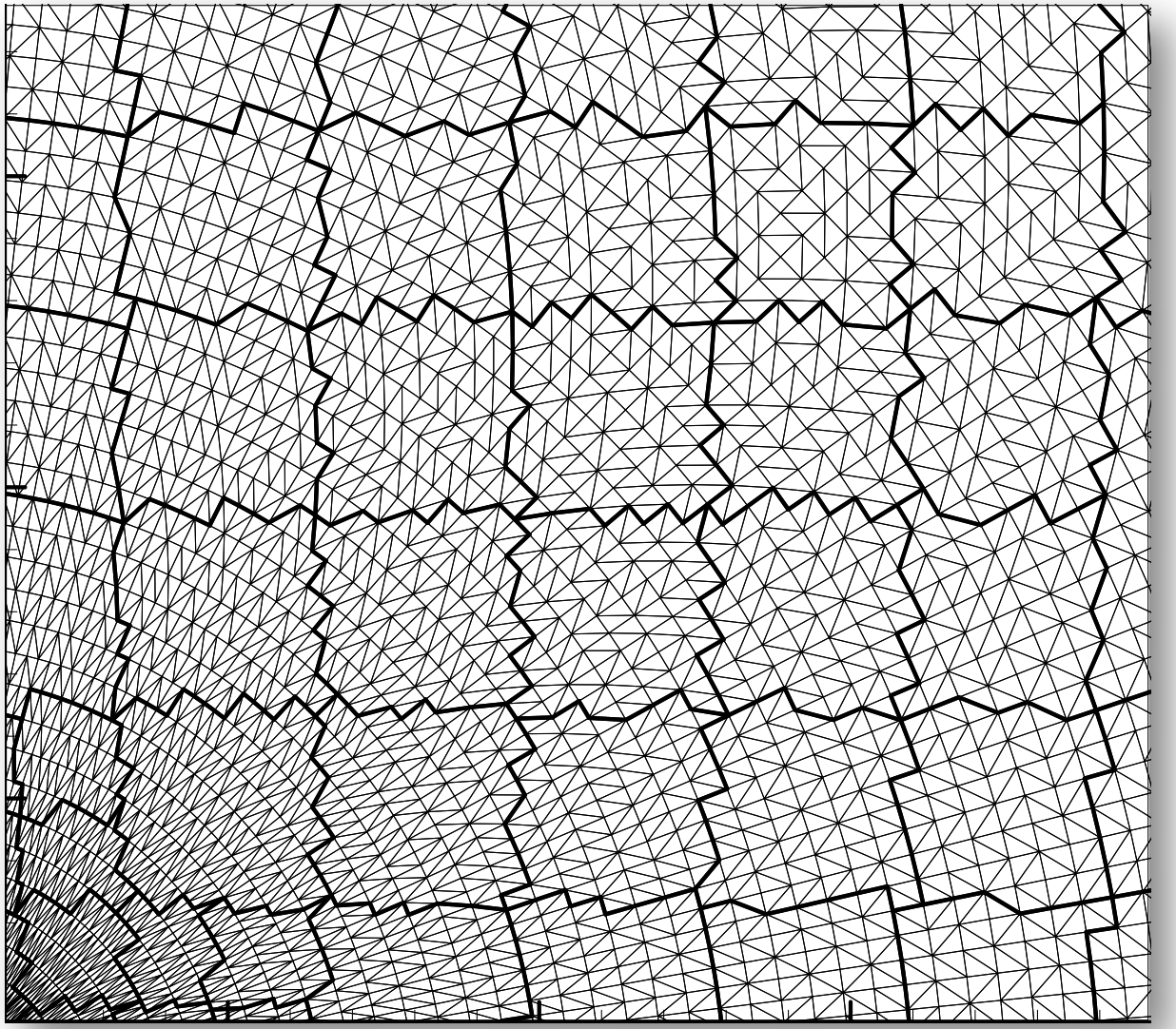


Data Required for Line 5/16 of total data

- Loading time reduced
- Memory requirements reduced

tecplot® Master the View

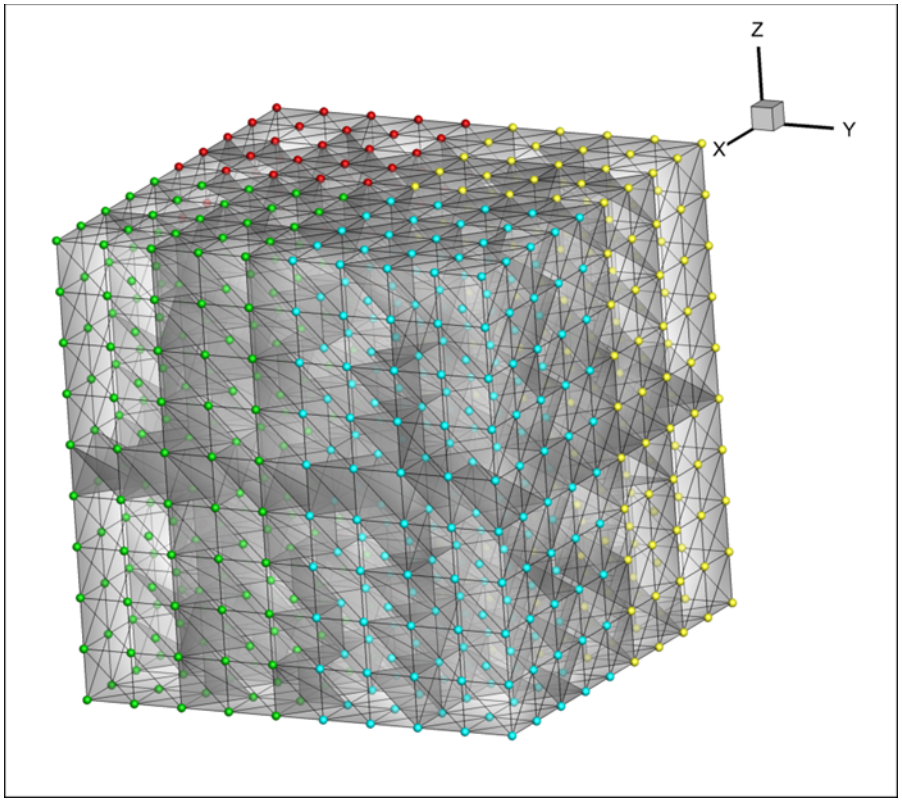
SZL Extended to Unstructured Data



tecplot® Master the View

3D Unstructured-Grid Domain Decomposition

Subdivision using Recursive Orthogonal Bisection



tecplot® Master the View

Progress Since 2012

- We were just starting in 2012, so there are nearly everything qualifies as progress, but will highlight four things:
  - Improved Tree Structure (indexing) for Queries
  - Interval Arithmetic for Queries of Implicit Variables (variables that are not yet computed functions of available variables)
  - File Data Compression
  - Parallel TecIO for writing from CFD codes

Progress: 256 Tree for Subzone Query

- Originally used classic interval tree, but this tree is smaller and more efficient
  - 256 cells/nodes per subzone
  - Each node of tree contains 256 spatially (x,y,z) related subzones
  - Query is O(log(N))

Progress: Interval Arithmetic to Query Based on Implicit Variable

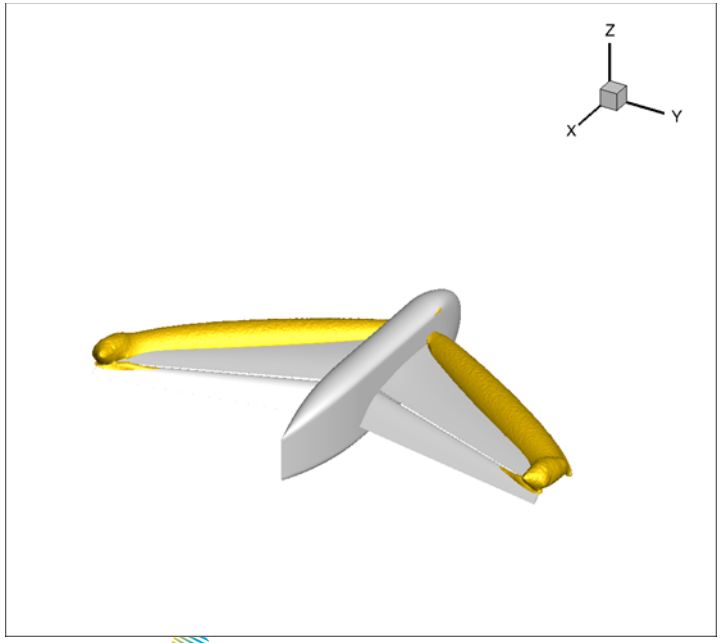
- How do you avoid the chicken-and-the-egg scenario:
  - Have conservative variables
  - Want isosurface of pressure (implicit variable – not in file)
  - Computing the pressure at each node to determine needed subzones violates the basic tenant of SZL (no O(N) operations)
- Solution: **Interval Arithmetic** computes a bound for the pressure range based on the ranges of the conservative variables

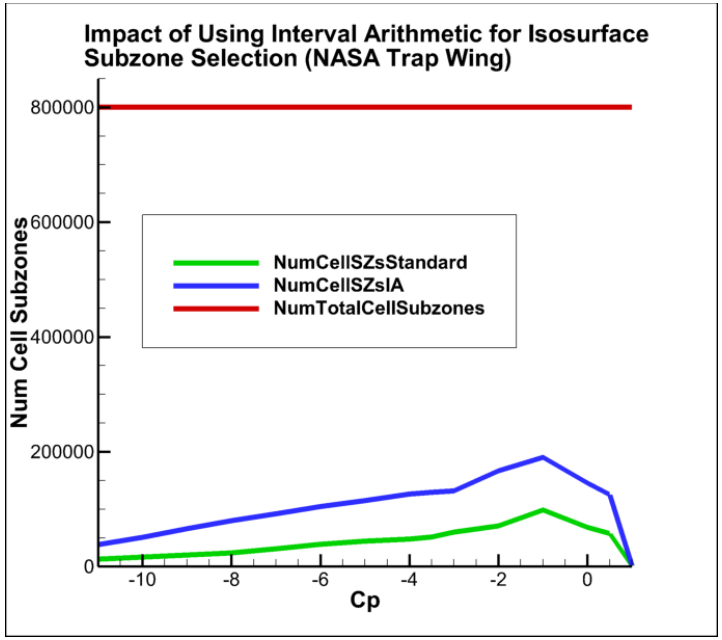
What is Interval Arithmetic?

- "…an approach to putting **bounds** on rounding errors … in mathematical computation…" (Wikipedia)
- Interval definition in our case, [min, max] (i.e. min and max of a variable in the subzone)
- Basic Operations:
  - [a,b] + [c,d] = [a+c,b+d]
  - [a,b] – [c,d] = [a-d,b-c]
  - [a,b] * [c,d] = [min(ac,ad,bc,bd),max(ac,ad,bc,bd)]
  - Etc.
- Interval arithmetic operations can be defined for nearly any function
- Use sequence of interval arithmetic operations to determine subzone [p_min,p_max] from the subzone [min,max] of the conservative variables

IA Example: Pressure Isosurface

- Tree contains [min,max] values of conservative variables
- Use interval arithmetic to estimate (bounds) [min,max] values of pressure for each subzone
- Only load data for those subzones containing the isosurface value





tecplot® Master the View

Loads more subzones, but still less than 20% of total.

Progress: Node-Map Compression

- For each cell subzone, replace standard node map by
  - a list of variable precision integers containing offset, O_{corner}^{cell}, into a list of referenced node subzones, S_n, and
  - a list of 8-bit integers containing the index, n_{corner}^{cell}, within that node subzone.

Example: 4 node sz's
4 32-bit integers

For Cell 1

$$S_1, S_2, \ldots$$

$$O_1^1, O_2^1, O_3^1, O_4^1, O_5^1, O_6^1, O_7^1, O_8^1$$
$$O_1^2, O_2^2, O_3^2, O_4^2, O_5^2, O_6^2, O_7^2, O_8^2$$
$$\ldots$$

2048 4-bit integers

$$O_1^{256}, O_2^{256}, O_3^{256}, O_4^{256}, O_5^{256}, O_6^{256}, O_7^{256}, O_8^{256}$$
$$n_1^1, n_2^1, n_3^1, n_4^1, n_5^1, n_6^1, n_7^1, n_8^1$$
$$n_1^2, n_2^2, n_3^2, n_4^2, n_5^2, n_6^2, n_7^2, n_8^2$$
$$\ldots$$

2048 8-bit integers

$$n_1^{256}, n_2^{256}, n_3^{256}, n_4^{256}, n_5^{256}, n_6^{256}, n_7^{256}, n_8^{256}$$

Total: 3088 Bytes
62% compression
Simple and fast!

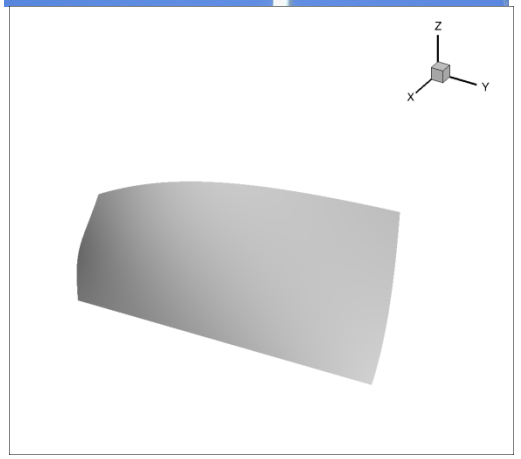
tecplot® Master the View

Progress: TecIO for writing from CFD codes

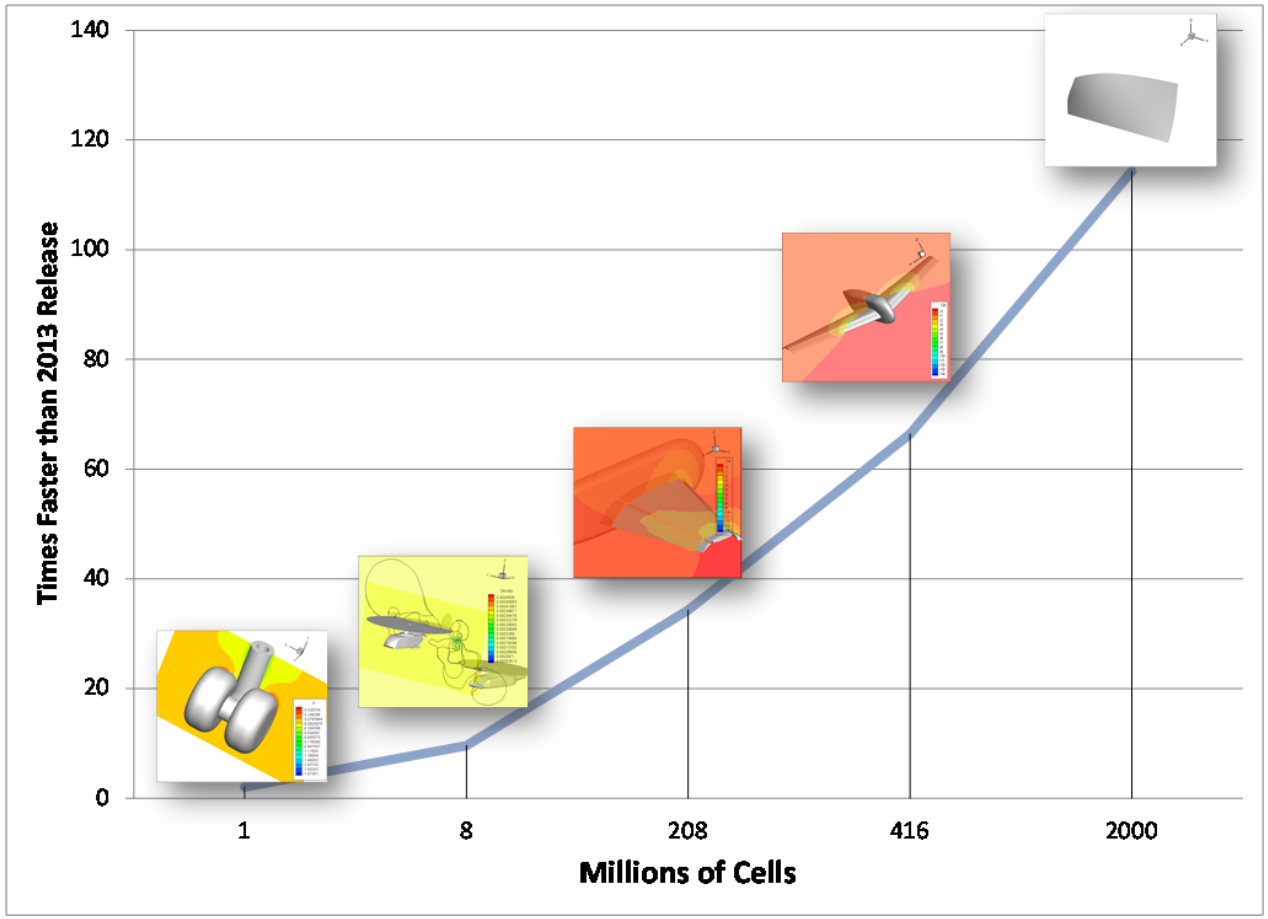
- Start with existing domain decomposition
- Further domain decomposition performed on each I/O node of HPC

Test Cases

- Performance scaling
  - Scaling up to 2 billion cells
- Unsteady wind-turbine analysis
  - Overflow results
- Grand Challenge: Synthetic dataset up to 1/3 Trillion cells



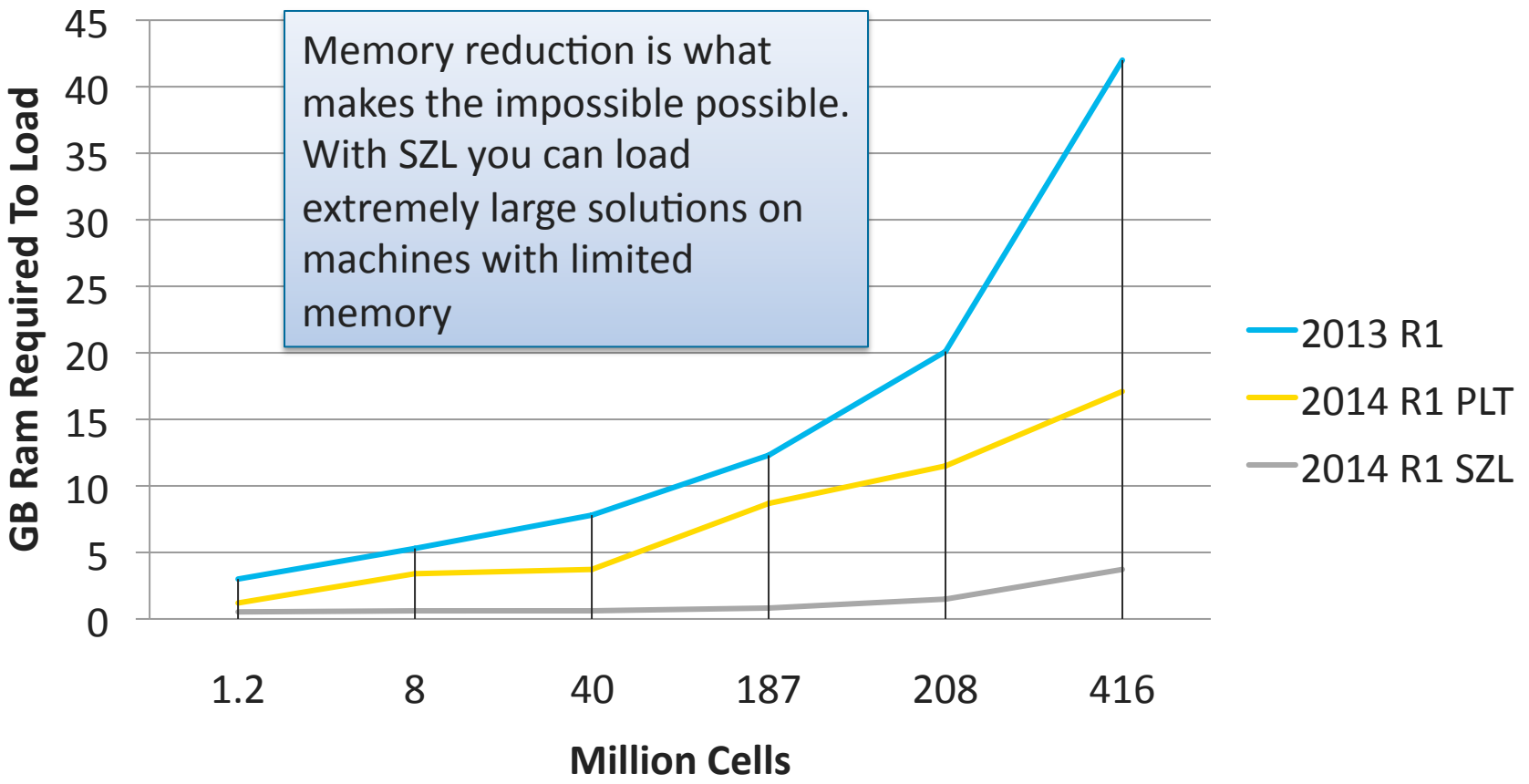
Data Processing Speedups With SZL



- 100% of the test show improved performance with SZL
- Results vary depending on specific data set, but all but the smallest test showed >10x speedups

tecplot® Master the View

Memory Reduction

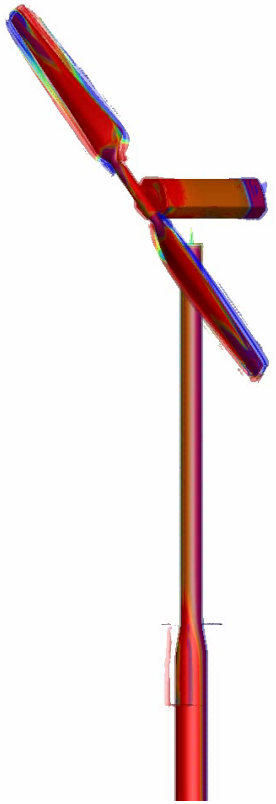


tecplot® Master the View

NREL Wind-Turbine Analysis

- 10m research turbine
- Acoustics (coupled to PSU WOPWOP)
- OVERFLOW 2.2
  - OVERFLOW-D mode
    - Domain Connectivity Function (DCF)
    - Geometry Manipulation Protocol (GMP)
  - 16 near-body blocks (2.6M points)
  - Adaptive Mesh Refinement
  - 2^{nd}-order differencing near-body
  - 4^{th}-order differencing off-body
- Results
  - 10 m/s aligned with turbine
  - 7152 time step (saved every 16 time steps)
  - Start: 30.5M nodes and 68 blocks
  - End: 260M nodes and 5600 blocks

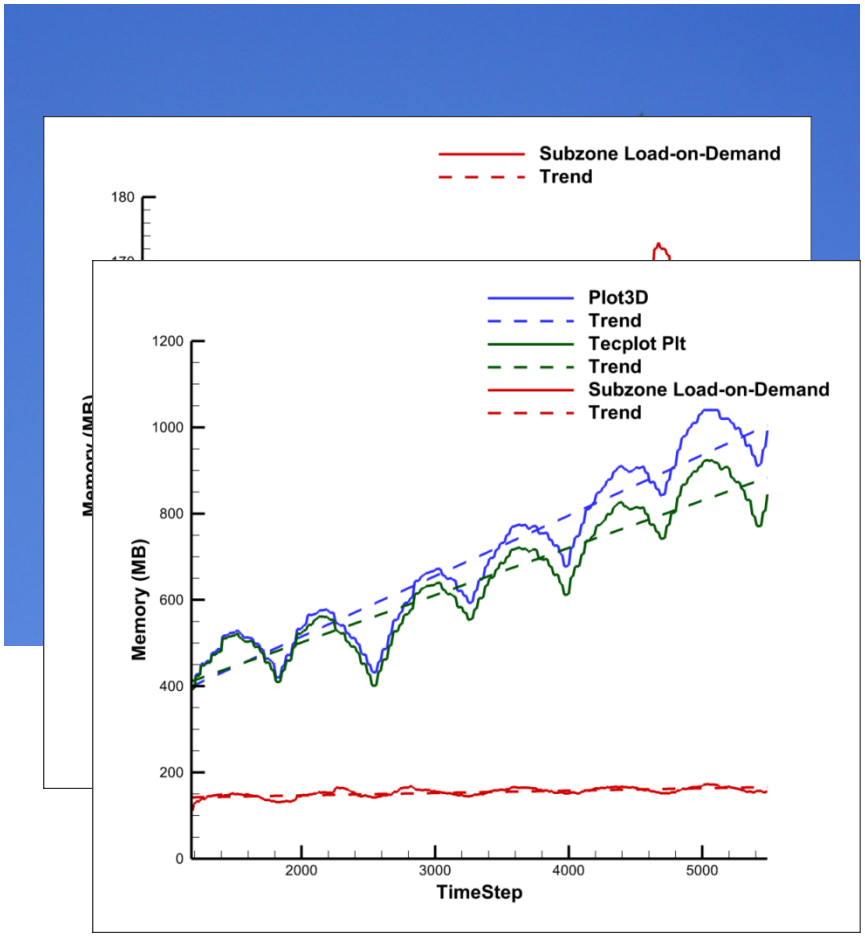
Animation of Wind Turbine Vorticity Magnitude

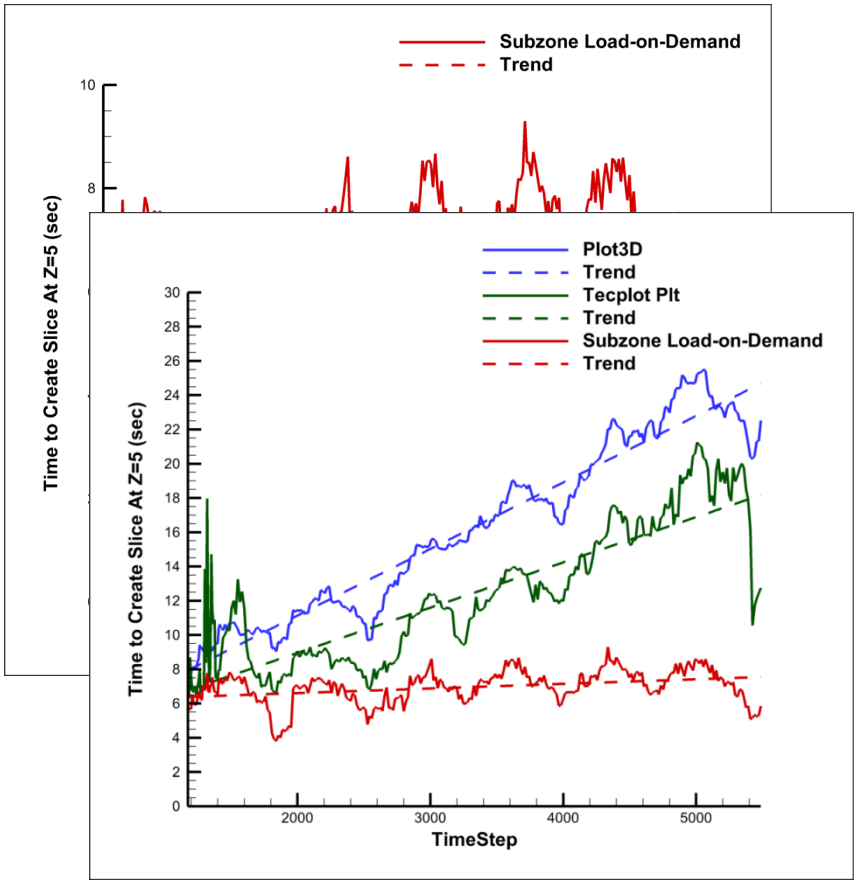


tecplot
Master the View

tecplot
Master the View

SZL Performance for Overset Grid

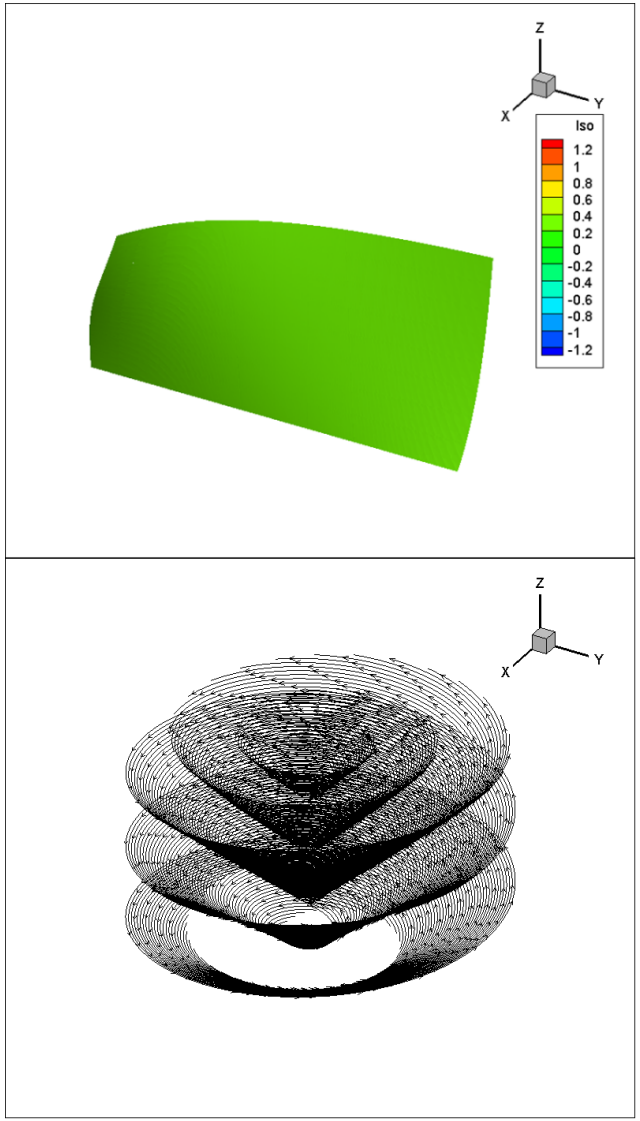




tecplot® Master the View

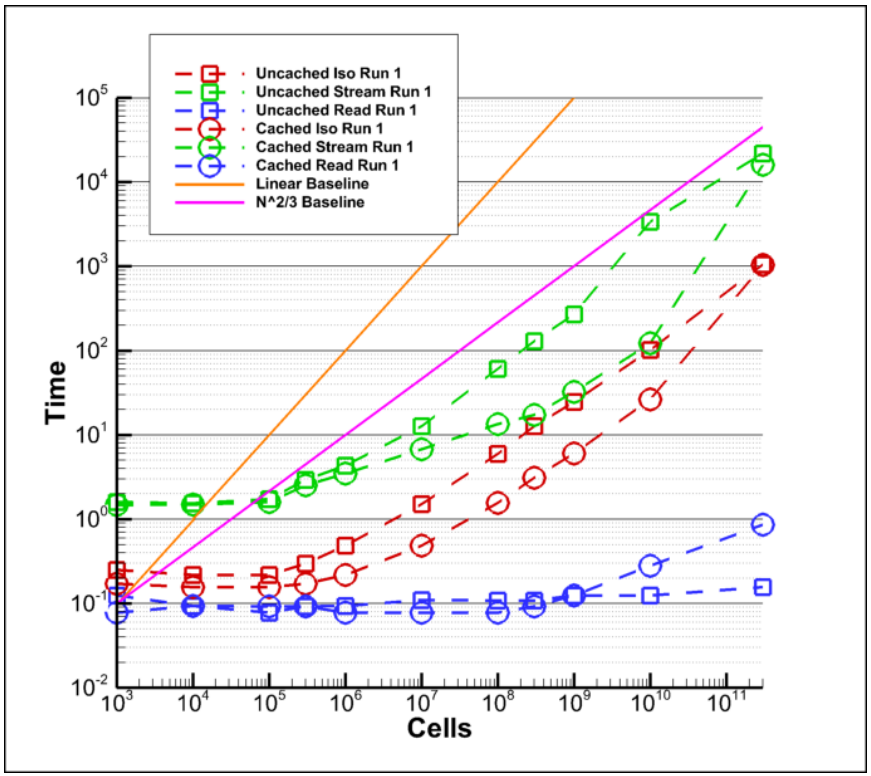
Grand Challenge Datasets

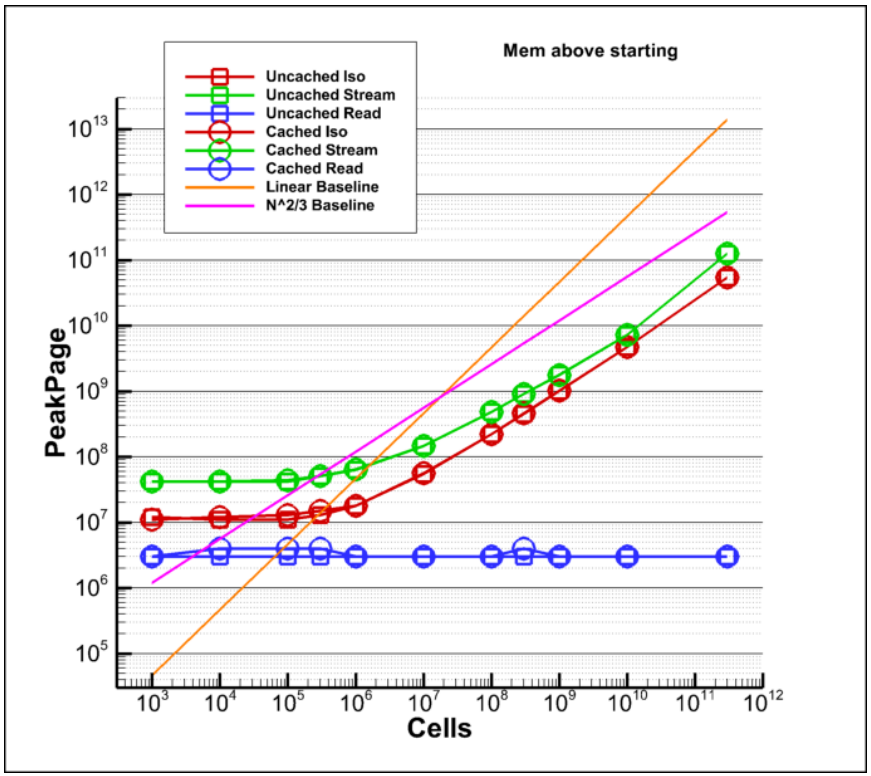
- Synthetic dataset
  - FE Tetrahedral
    - Isosurface variable: simple polynomial variation of a scalar
    - Velocities: straight Rankine vortex with constant axial flow
  - Size Scaling up to 1/3 Trillion Cells (to date)
  - Test 1: Isosurface
  - Test 2: Set of 64 streamtraces



tecplot® Master the View

Grand Challenge Results





tecplot® Master the View

Conclusions

- Dramatic reduction in memory requirements
  - Scaling for isosurface and slices is $O(N^{2/3})$ - critical for maintaining performance into the future
  - Scaling for a individual streamtrace is $O(N^{1/3})$
- Significant improvements in speed for most cases
  - Up to 100 times faster for real datasets
  - 3 times faster for overset data with large number of zones
- Significant progress in SZL algorithm
  - Tree structures for optimized queries
  - Implicit Variables
  - Data compression
  - Efficient writing of SZL files from parallel (MPI) CFD codes

Future Work

- SZL support for remaining features in Tecplot
  - Implicit derivatives (vorticity, lambda-2, …)
  - FE-boundary extraction
- Optimize for large number of zones (Overset)
  - Eliminate O(M) operations (M=number of zones)
  - Improve handling of blanking
- Improve performance of subzone server for remote data
- Further speed improvements (factor of 3-10 faster)
  - Further threading
  - Eliminating remaining O(N) operations
  - Apply principles to non-SZL files???

Questions?

If you have questions about this technology, please talk with Scott (s.imlay@tecplot.com) or Durrell (d.rittenberg.@tecplot.com).

tecplot® Master the View

Common Questions and Answers

- We have a high-performance parallel file system, do we still need SZLOD?
  - YES. A high-performance parallel file system will delay the problem, but disk read performance will still eventually be the bottleneck.
- How does this compare to the client-server model used by ____?
  - The current client-server models minimize the effect of slow networks, but do nothing to mitigate the effect of slow disk drives. Slow networks are the bottleneck of the past, slow hard disks are the bottleneck of the future.
  - We are working on a sub-zone server to address customers client server needs.