Recent Advances in Overcoming the Red Shift for CFD Simulation Analytics

49th HPC User Forum, Tucson AZ

Scott T. Imlay

Chief Technology Officer, Tecplot Inc.
Outline

• Hardware Trends
• CFD Usage Trends
• Analysis – affect of trends on visualization & analysis pipeline
• Proposed solutions
• Results
Tecplot

• Founded in 1981 by two former Boeing employees (Mike Peery & Don Roberts)
• First 15 we developed CFD codes
• Now focus on post-processing analysis and visualization
• 40,000 users world wide (60% domestic)
• On-going performance initiative
Red Shift

• Difference in performance improvement between CPU cores and the components feeding them data
  – Primarily interested in Disk I/O
Driving Force is Moore’s Law

Supercomputer performance is tracking with Moore’s Law
Hard-drive Load Times Dominate

- Disk Capacity is doubling every 12 months
- Disk read data transfer rate doubling every 36 months
CFD Dataset Size Growing with Moore’s Law

- Wide range in length scales
- Resolution of grid (# of grid points) constrained by computer performance
Parametric CFD Analysis

Highly-Dimensional Collection of Data:

- Aero Database Development
  - Determine aerodynamic characteristics over subset of flight envelope
    - Mission space: Speeds and angles of flight
    - Configuration space: Control positions, etc.
    - CFD data space: x, y, z, perhaps time
- Optimization or Robust Engineering
  - Additional parameters for geometry
- Verification & Validation
  - Evaluate codes, code parameters, subscale models, etc.

Impact:

- Multiple CFD runs in each dimension
- 100s or 1000s of CFD datasets generated over months or years – many TeraBytes of data
- Simulation Analytics is the simultaneous analysis and visualization of all these simulation runs
  - Design space (highly dimensional
  - Physical space

From AIAA 2004-5076
Tecplot Chorus For Simulation Analytics

Evaluating overall system performance and allowing engineers to compare results quickly.
Ramification of Simulation Analytics

• Operations of enormous amounts of data
  – Example: Aero database development
    • Thousands of 100M cell CFD solutions
  – Some operations require data from all sources to be analyzed simultaneously
    • If no clever, must work through equivalent of 100B cells

• Large data performance issues become dramatically worse
Data Processing Pipeline

Data Files → Bandwidth → CPU/GPU

- Double every 36 months
- Double every 16 months
- Double every 18 months

Data IO is the current rate determining step in the visualization pipeline.
Consequence of Red Shift

- Current visualization architectures will perform worse as time goes on!
Overcoming Data Transfer Bottleneck
Popular Approaches in Industry

• Hardware/System Improvements
  – Parallel file systems (delays problem, but can’t outgrow Moore’s law by adding spindles)
  – New types of memory
    • SSD (probably expensive for many of our customers)
    • Holographic memory, etc. (not soon)

• In Situ visualization
  – Link libraries into CFD code to extract desired data or images (Don’t save volume data)
  – Circumvents the disk transfer rate bottleneck
  – What about aggregations and data mining?

• Parallel visualization
  – Doesn’t entirely solve disk transfer rate problem
  – May help some if it uses efficient parallel data reads
  – Red Shift doesn’t need more compute power!
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 (SZLoD)
  – Save indexed volume data file
  – Load only the data you need (Lazy Loading)
  – Related work
    • Out-of-Core algorithms of the 1990’s
    • Field Encapsulation library of Patrick Moran at NASA Ames
How Does SZLoD 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
The indexed decomposition can be extended to 3D for iso-surfaces, slices and streamtraces.
SZLoD Extended to Unstructured Data

Subdivision using Recursive Orthogonal Bisection
Indexing for Subzone Selection - Interval Tree

Binary tree of intervals (value ranges)
- Return all intervals that contain a specified value of the variable
- 255 cells per subzone
- Query is \( O(\log(N)) \)

<table>
<thead>
<tr>
<th>Grid Size (Cells)</th>
<th>Size (subzones)</th>
<th>Query (no tree)</th>
<th>Query (tree)</th>
<th>Tree file size</th>
</tr>
</thead>
<tbody>
<tr>
<td>1B</td>
<td>4M</td>
<td>17ms</td>
<td>0.12ms</td>
<td>62.8MB</td>
</tr>
<tr>
<td>10B</td>
<td>40M</td>
<td>160ms</td>
<td>1.4ms</td>
<td>620MB</td>
</tr>
</tbody>
</table>
Test Cases

• Synthetic test dataset
  – Scaling up to a billion cells
• Transport aircraft
  – 187 Million cell finite-element grid
• Unsteady wind-turbine analysis
  – Overflow results
• NASA Trapezoidal Wing (High Lift Prediction Workshop)
  – 204 Million cell finite-element grid
Scaling of Subzone LOD with Dataset Size

• Overcoming Red Shift
  – Need sub-linear scaling with number of cells
  – SZLoD scales $O(N^{2/3})$
FE Transport Aircraft – Slice

Slice Memory Requirement Ratio

Slice Timing Ratio

187M Cells
Transport Aircraft
Landing Configuration
FE Transport Aircraft – Streamtrace

- Tecplot
  - 170 sec
  - 16 GB

- SZLOD
  - 2.2 sec
  - 1.3 GB max
  - 0.7 GB resting
Animation of Wind Turbine Vorticity Magnitude
SZLoD Performance for Overset Grid
Full Trap Wing Results – Isosurface

• Generate Isosurface, $C_p = -2$
  - 408M FEBrick cells in volume
  - 4.7M triangles in isosurface
  - 16x faster than standard Tecplot

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Time (sec)</th>
<th>Peak Mem (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Tecplot</td>
<td>700</td>
<td>49</td>
</tr>
<tr>
<td>Subzone Load-on-Demand</td>
<td>43</td>
<td>2.4</td>
</tr>
</tbody>
</table>
Half Trap Wing Results – Slice

- Generate Slice at y=100
  - 94x faster than standard Tecplot
  - 540x faster than single-threaded Tecplot
  - 55x less memory

<table>
<thead>
<tr>
<th>Algorithm Used</th>
<th>Time (sec)</th>
<th>Peak Mem (GB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Tecplot</td>
<td>222</td>
<td>20.5</td>
</tr>
<tr>
<td>Single-Threaded Tecplot</td>
<td>1279</td>
<td>20.4</td>
</tr>
<tr>
<td>Subzone Load-on-Demand</td>
<td>2.36</td>
<td>0.366</td>
</tr>
</tbody>
</table>
Conclusions

• Dramatic reduction in memory requirements
  ─ Factor of 4 to 50 less memory used
  ─ Scaling for isosurface and slices is $O(N^{2/3})$ - critical for maintaining performance into the future
  ─ Scaling for a streamtrace is $O(N^{1/3})$

• Significant improvements in speed for most cases
  ─ 15 to 120 times faster for synthetic data and transport aircraft
  ─ 3 times faster for overset data with large number of zones

• Similar benefits when network bandwidth is bottleneck

• Downside
  ─ Speedups depend on using new file format (but you can still get memory reductions with native files)
Questions?

If you are interested in testing this technology, please talk with Scott (s.imlay@tecplot.com)