Architectural Challenges and Solutions for Petascale Visualization and Analysis

Hank Childs
Lawrence Livermore National Laboratory

June 27, 2007
Overview

• Large scale data creates two incredible challenges: **scale** and **complexity**

• **Petascale** is not “business as usual”
  – Current trajectory for terascale postprocessing will be cost prohibitive at the petascale
  – We will need “smart” techniques in production environments

• We can reduce **complexity** through diverse use cases that go beyond scientific visualization
  – This is also responsive to user requests
  – The cost of entry for petascale postprocessing is high; economies are achieved by targeting many use cases simultaneously

www.vacet.org
Current modes of production postprocessing (1)

- SC and PP* cluster share disk
- PP cluster has good I/O access
- SC runs lightweight OS; PP cluster runs Linux
- Graphics cards on PP cluster

*: PP = Postprocessing
Current modes of production postprocessing (2)

- Simulation, processing both done on purple
- Simulation writes to disk, PP* job reads from disk
- SC runs full OS (AIX)
- No graphics cards

*: PP = Postprocessing
These modes of postprocessing have worked well at the terascale.

- Rayleigh Taylor instability by MIRANDA code
- 27 billion elements
- Run on ASC BG/L
- Visualized on gauss
Our current trajectory for terascale postprocessing will be cost-prohibitive at the petascale.

ASC BG/L
360 TF
5 years: 5 PF

Current software strategy, “pure parallelism”:
- Read in all of the data
- Store in primary memory
- Run algorithms on data

Gauss
512 procs
$1-$2M
5000? procs
$15M (!!!)

*: PP = Postprocessing
Co-opting the SC is also problematic at the petascale.

- Lightweight OS’s present challenges
- We aren’t compute bound; multi-core has limited value-added, need to use more of the machine at petascale
- Use cases are “bursty” – do we want the SC sitting idle while someone studies results?

*: PP = Postprocessing
Scale challenge: petascale postprocessing mandates software investments beyond pure parallelism.

- Research has established viable, effective alternatives to pure parallelism
  - Out of core processing
  - In situ processing
  - Multi-resolution techniques
- These techniques are complex → the barriers to entry have risen
Complexity challenge: we must consider many diverse use cases.

• **Use cases:**
  – Data exploration
  – Quantitative analysis
  – Visual debugging
  – Presentation graphics
  – Comparative analysis

• Each use case lends itself to certain processing techniques, but not others

• By considering all of these use cases:
  – respond to demands of user community
  – creates a synergy, from both developer and customer perspectives
Addressing the complexity challenge: visualizing large data is necessary, not sufficient

27B elements

1B elements

Thank you, Wes Bethel!

Mesh-Based Simulation Data

www.vacet.org
A more complete analysis included identifying and enumerating bubbles and spikes.

- Meat grinder analogy: we need to spend our time turning the crank, but we also need to spend our time looking for the right meat.
- We can’t focus solely on visualization, also need analytics.

Credit: Timo Bremer, Valerio Pascucci, and more!
See poster tonight!

www.vacet.org
The synergy between the use cases is motivated by this example of combustion analysis.

Evaluate location of a flame front (5 triangles shown, but think millions)

Evaluate -velocity

Construct surface at previous time slice

Evaluate -velocity (again)

Repeat many times

Construct surface at previous time slice (again)

Create wedge mesh
The synergy between the use cases is motivated by this example of combustion analysis.

Partition the wedges based on the triangle they originate from.

Integrate mass fraction over a wedge, then divide by surface area

Calculate joint PDF of ratios.

→ Produced 1D curve, done “real science”
Each of these operations is turned into modules in an interoperable framework.

- Evaluate location of a flame front
- Evaluate velocity from a given time slice onto the current surface
- Create a new surface based on the velocity
- Isosurface
- Cross-mesh field evaluation
- Displacement
- Integration, ratio, PDF
The modules used span our use cases.

Data exploration
Comparative analysis
Visual debugging
Presentation graphics
Quantitative analysis

Isosurface
Cross-mesh field evaluation
Displacement
Integration, ratio, PDF

Other economies include a flexible, extensible infrastructure for large data, file readers, data model understanding, etc.
Designing a solution to the petascale problem with diverse use cases…

• No processing technique is a panacea
  – None can support every use case (with low cost)

• **Vision 1**: when possible, allow for functionality to be deployed using a variety of techniques
  – i.e. adapt to and leverage the available resources

• **Vision 2**: provide a suite of processing techniques through a single system
  – For users
  – For developers
We will construct a table to inform our path.

<table>
<thead>
<tr>
<th></th>
<th>Pure Parallelism</th>
<th>In Situ</th>
<th>Multires</th>
<th>Out-of-core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Exploration</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Quantitative Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Advanced</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Physics-based</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Comparative Analysis</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Image-Based</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Data-Level</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Topological</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Individual Queries</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Aggregate Queries</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Moviemaking</td>
<td>?</td>
<td>?</td>
<td>?</td>
<td>?</td>
</tr>
<tr>
<td>Presentation-oriented</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exploration-oriented</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
There are fundamental requirements for some use cases that cannot be met by some smart techniques.

- **Idle time**
  - Doesn’t fit: in situ
  - Required by: data exploration, visual debugging

- **Interactivity**
  - Doesn’t fit: out-of-core
  - Required by: data exploration, some visual debugging

- **Processing data in its full resolution and its native form**
  - Doesn’t fit: multi-resolution techniques
  - Required by: some debugging, quantitative analysis, comparative analysis, some moviemaking

- **Processing multiple data sets**
  - Doesn’t fit: in situ
  - Required by: comparative analysis

This is described in gross terms; the paper has more details.
<table>
<thead>
<tr>
<th></th>
<th>Pure Parallelism</th>
<th>In Situ</th>
<th>Multires</th>
<th>Out-of-core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Exploration</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Quantitative Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Advanced</td>
<td>Yes</td>
<td>Sometimes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Physics-based</td>
<td>Yes</td>
<td>Sometimes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Comparative Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image-Based</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data-Level</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Topological</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Visual Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual Queries</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Aggregate Queries</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Moviemaking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation-oriented</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Exploration-oriented</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Not all processing techniques are equal

- In situ
- Multi-resolution
- Pure parallelism
- Out-of-core

Acceptable
Cost prohibitive

Computer-cost too high
Runtime-cost too high
<table>
<thead>
<tr>
<th>Feature</th>
<th>Pure Parallelism</th>
<th>In Situ</th>
<th>Multires</th>
<th>Out-of-core</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Exploration</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Quantitative Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Advanced</td>
<td>Yes</td>
<td>Sometimes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Physics-based</td>
<td>Yes</td>
<td>Sometimes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Comparative Analysis</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image-Based</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data-Level</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Topological</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Visual Debugging</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual Queries</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Aggregate Queries</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Moviemaking</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation-oriented</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Exploration-oriented</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Vision: a software infrastructure that enables economy in software development

- Software infrastructure handles the details of how data is processed.
- Algorithm development is done independent of data processing technique.
Is it possible to have a software infrastructure that enables “processing technique indifference”?

- **VisIt:**
  - Full implementation of pure parallelism
  - Implementation of out-of-core
    - Exception: algorithms that require all data to be in memory fall back to pure parallelism
  - Implementation of in situ
    - Problem: makes a copy of data … memory “tax” too high
  - No multi-resolution implementation
    - No known technical barriers
    - (AMR DATA ALREADY SUPPORTED IN FULL)
<table>
<thead>
<tr>
<th></th>
<th>Pure Parallelism</th>
<th>In Situ</th>
<th>Multires</th>
<th>Out-of-core</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data Exploration</strong></td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td><strong>Quantitative Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Basic</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Advanced</td>
<td>Yes</td>
<td>Sometimes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Physics-based</td>
<td>Yes</td>
<td>Sometimes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Comparative Analysis</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image-Based</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Data-Level</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Topological</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Visual Debugging</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Individual Queries</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Aggregate Queries</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>Moviemaking</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Presentation-oriented</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Exploration-oriented</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Summary

• Petascale postprocessing is not “business as usual”
  – To move forward, we need smart techniques applied to varied use cases
• Some needed assets are in place, some are still being worked on.
• VACET is working on closing these gaps.