VACET

# Architectural Challenges and Solutions for Petascale Visualization and Analysis

Hank Childs

Lawrence Livermore National Laboratory

June 27, 2007

Work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

Lawrence Livermore National Laboratory, P.O. Box 808, Livermore, Ca, 94551

www.vacet.org

This is presentation UCRL-PRES-232039

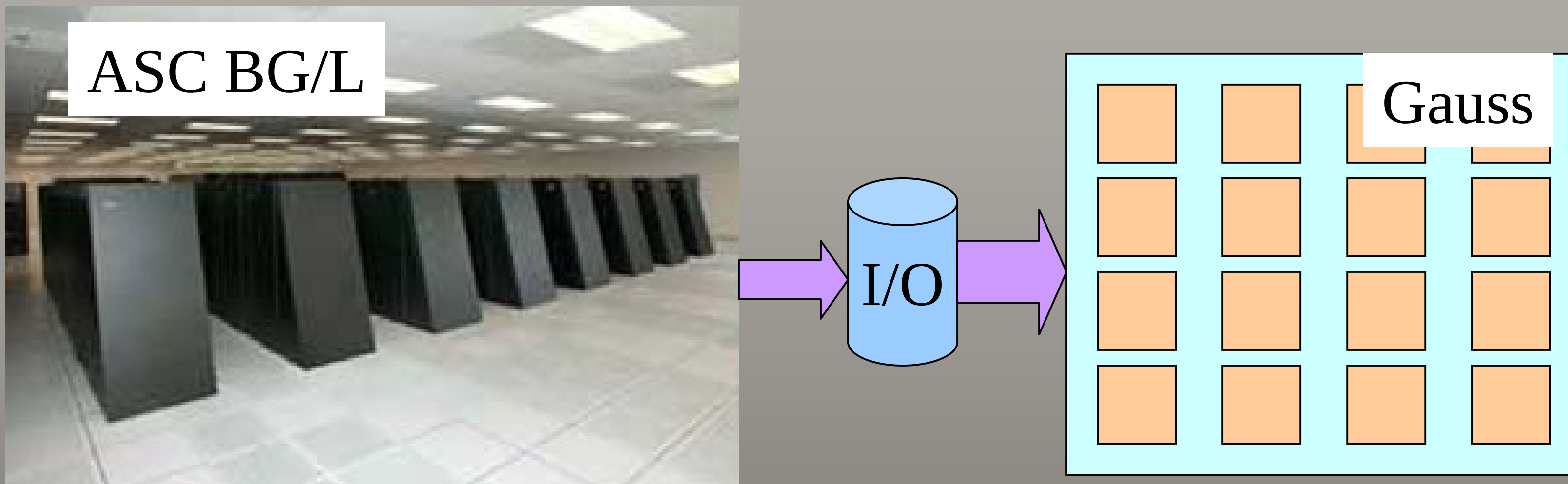
# Overview

- 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

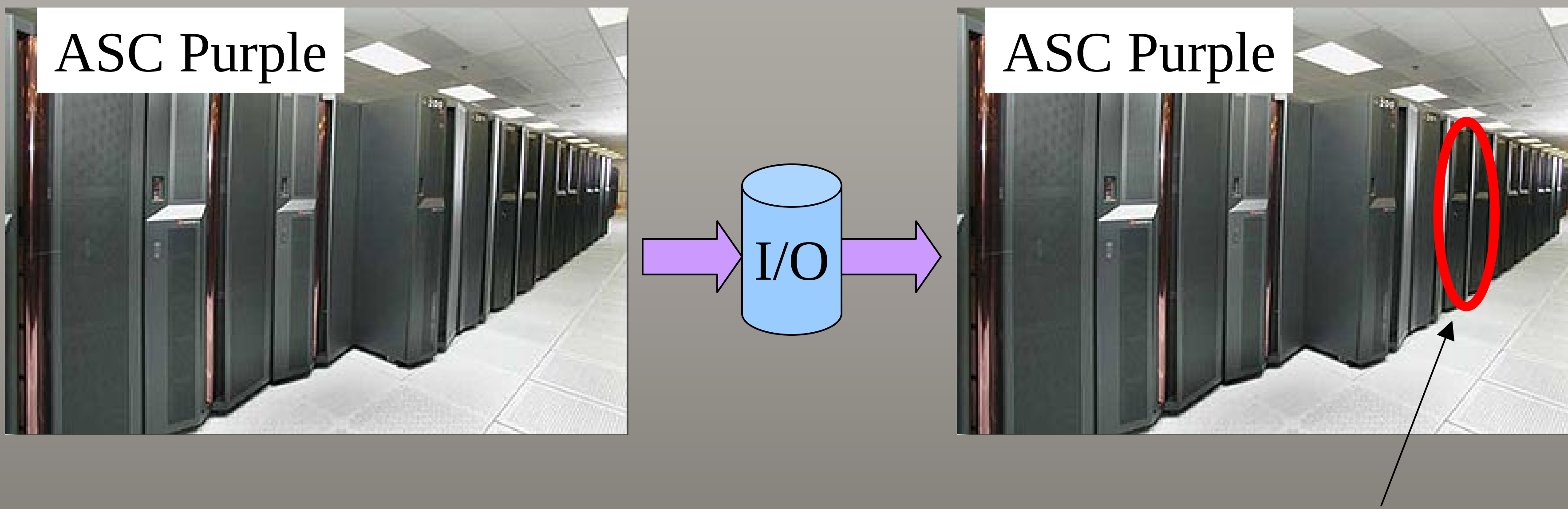
# Current modes of production postprocessing (1)




- 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

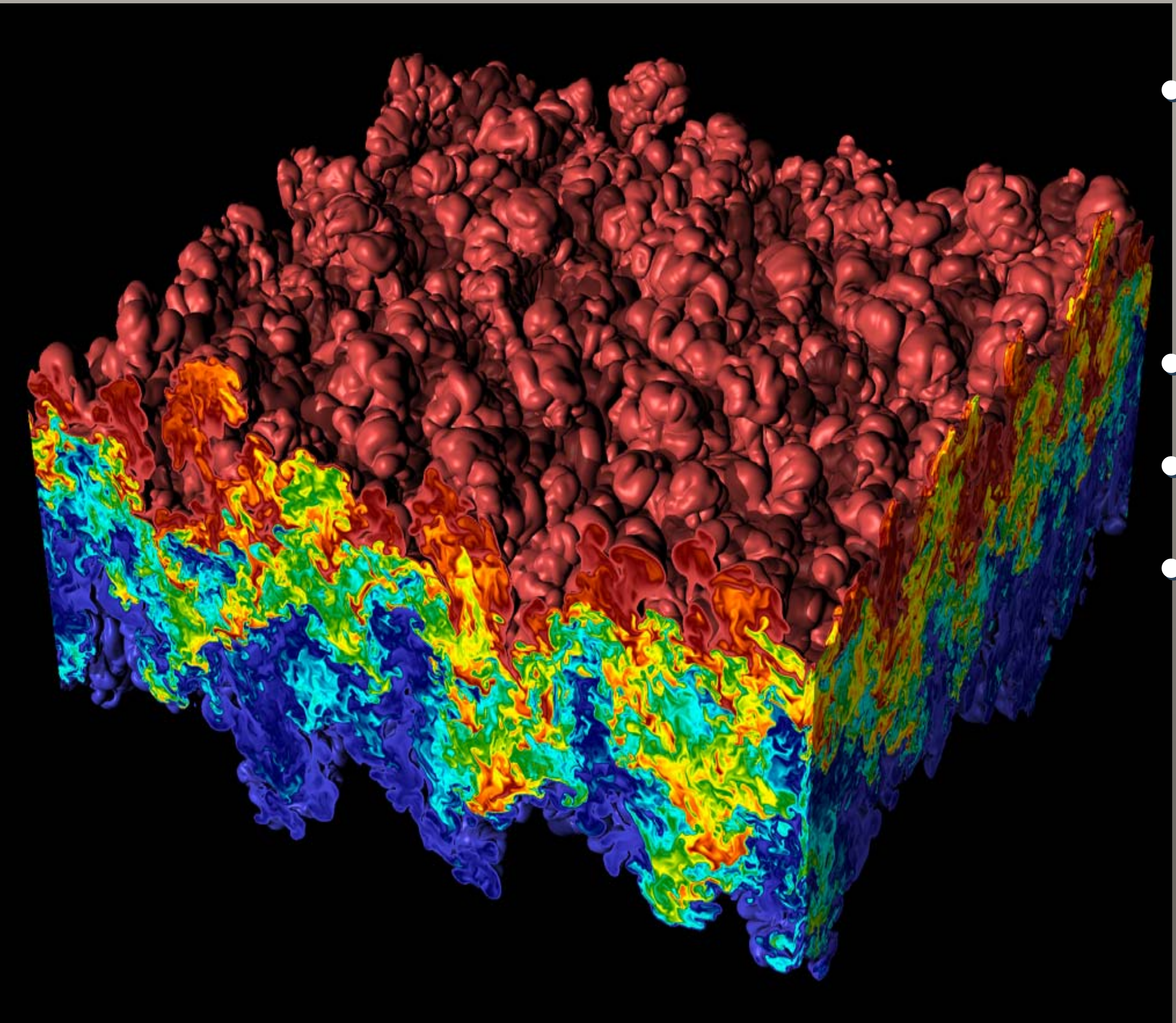
# Current modes of production postprocessing (2)




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

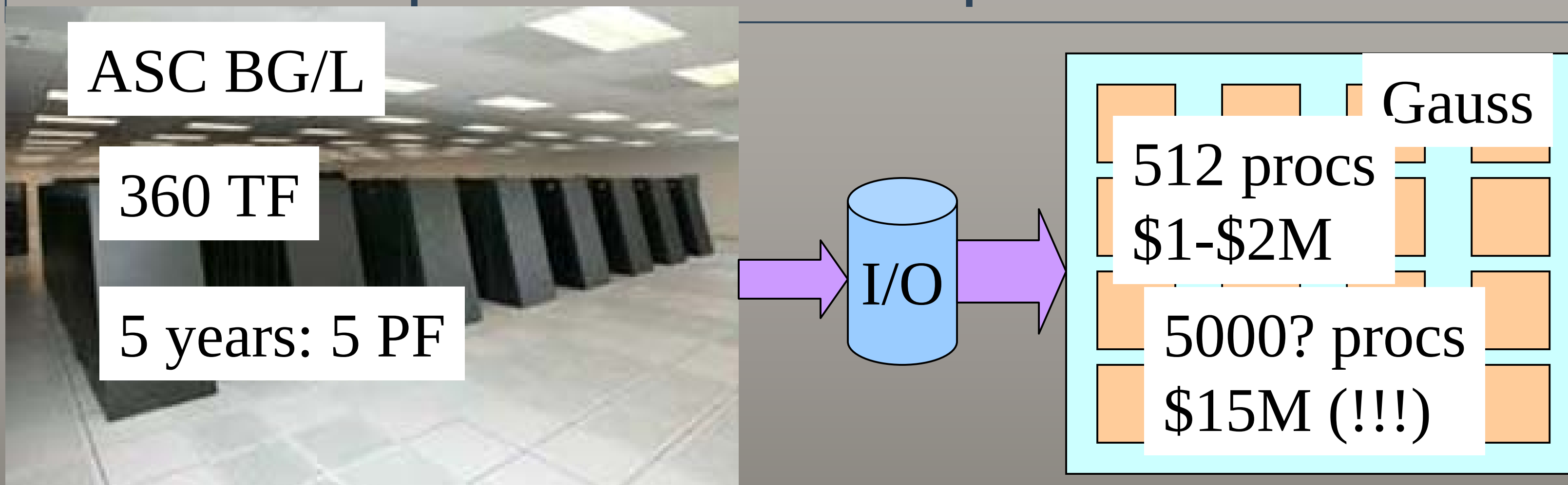
*: PP = Postprocessing

# 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

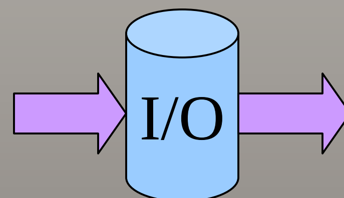
## Our current trajectory for terascale postprocesing will be cost-prohibitive at the petascale.




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

*: PP = Postprocessing

# Co-opting the SC is also problematic at the petascale.



Portion of purple for PP

- 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

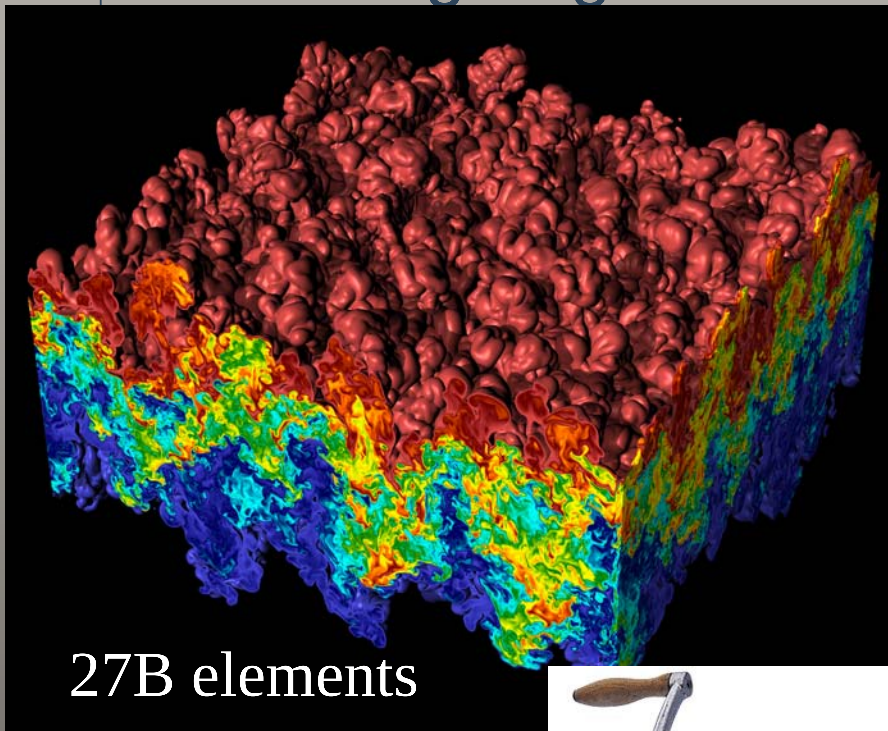
## Scale challenge: petascale postprocessing mandates software investments beyond pure parallelism.

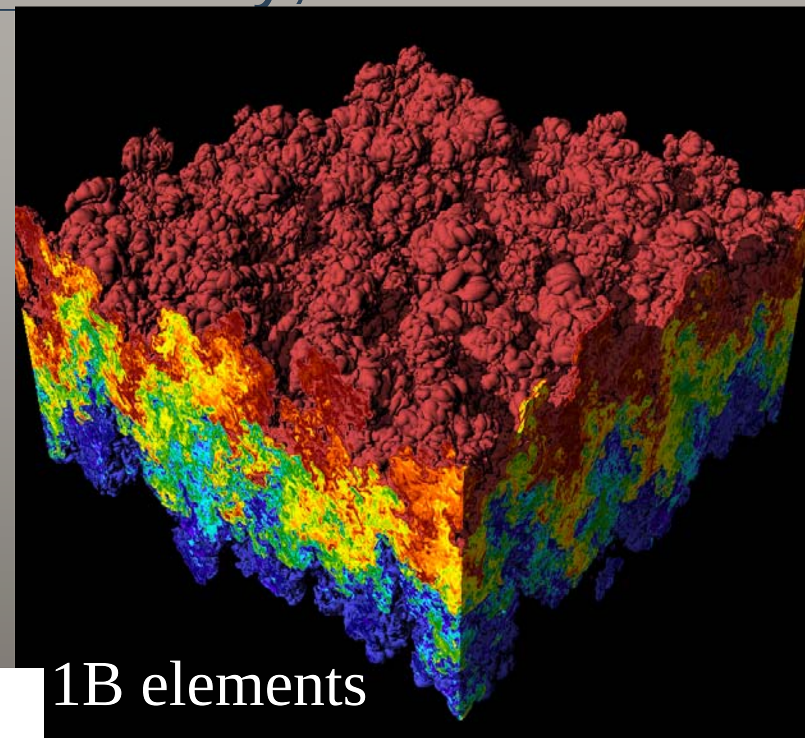
- Research has established viable, effective alternatives to pure parallelism
  - Out of core processing
  - In situ processing
  - Multi-resolution techniques
- These techniques are complex → the barriers to entry have risen

# 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

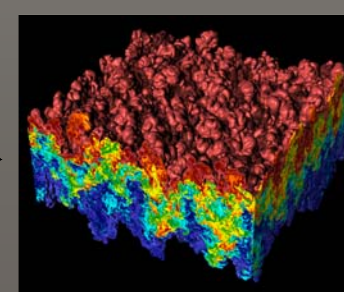
# Addressing the complexity challenge: visualizing large data is necessary, not sufficient



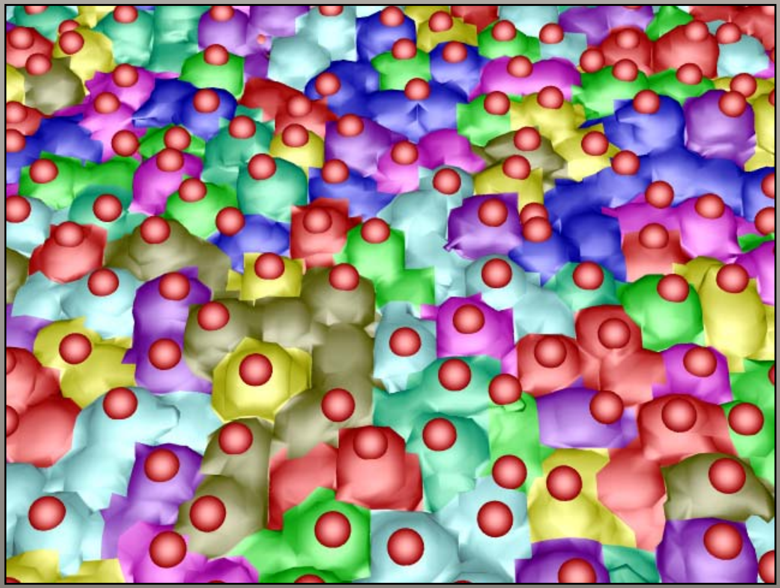



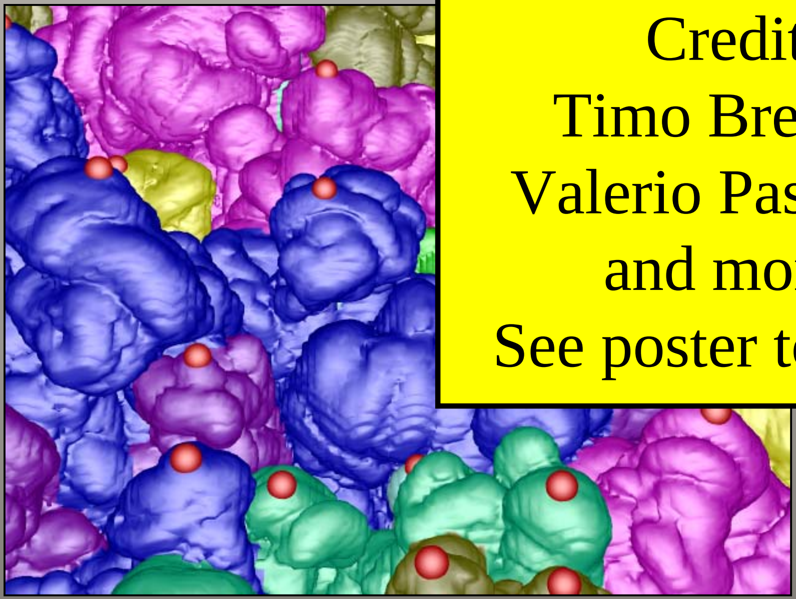

Mesh-Based Simulation Data

Thank you, Wes Bethel!



# A more complete analysis included identifying and enumerating bubbles and spikes.

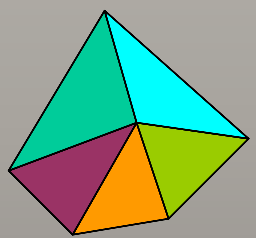




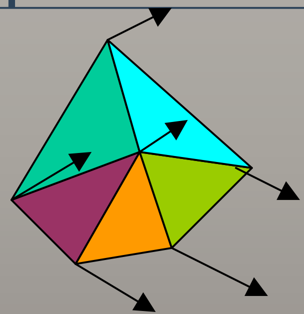
Credit:
Timo Bremer,
Valerio Pascucci,
and more!
See poster tonight!

- 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.

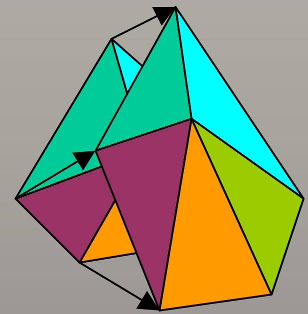
# The synergy between the use cases is motivated by this example of combustion analysis.



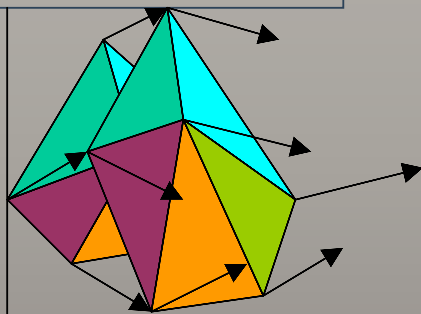
Evaluate location of a flame front (5 triangles shown, but think millions)



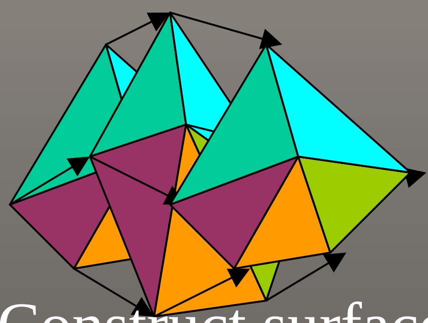
Evaluate -velocity



Construct surface at previous time slice

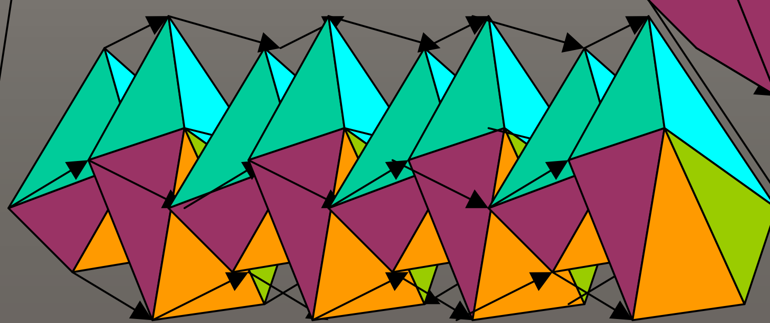


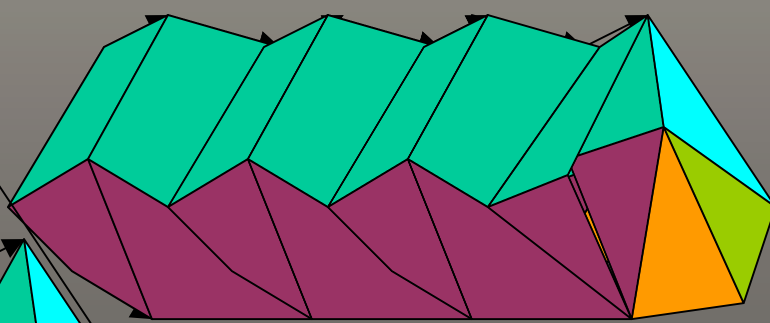
Evaluate -velocity (again)



Construct surface at previous time slice (again)

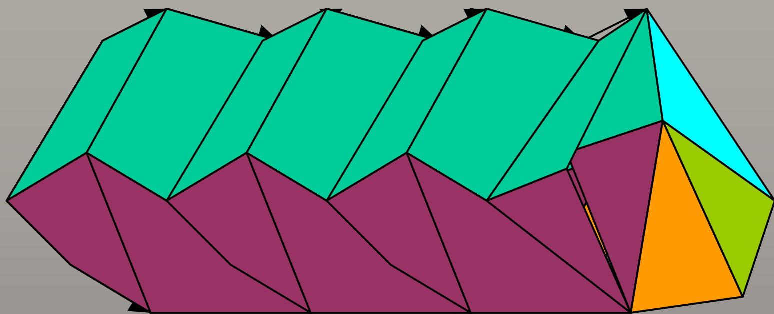
Repeat many times

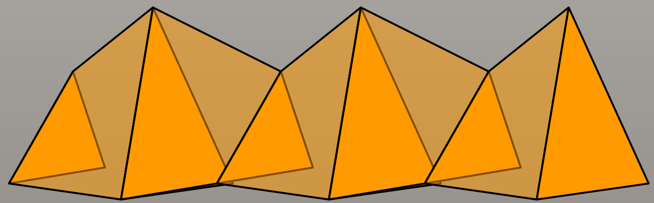




Create wedge mesh

# The synergy between the use cases is motivated by this example of combustion analysis.



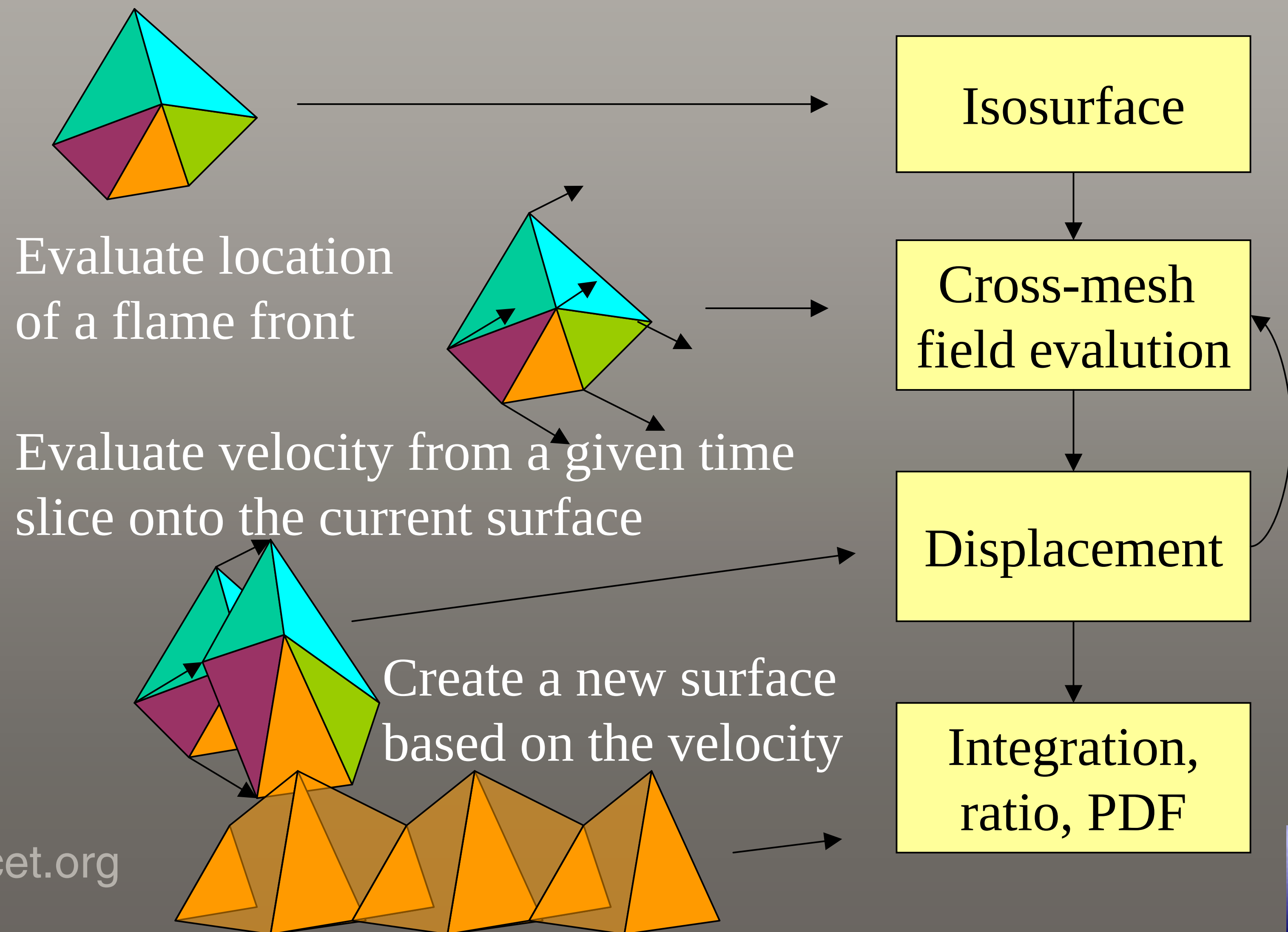


Partition the wedges based on the triangle they originate from.

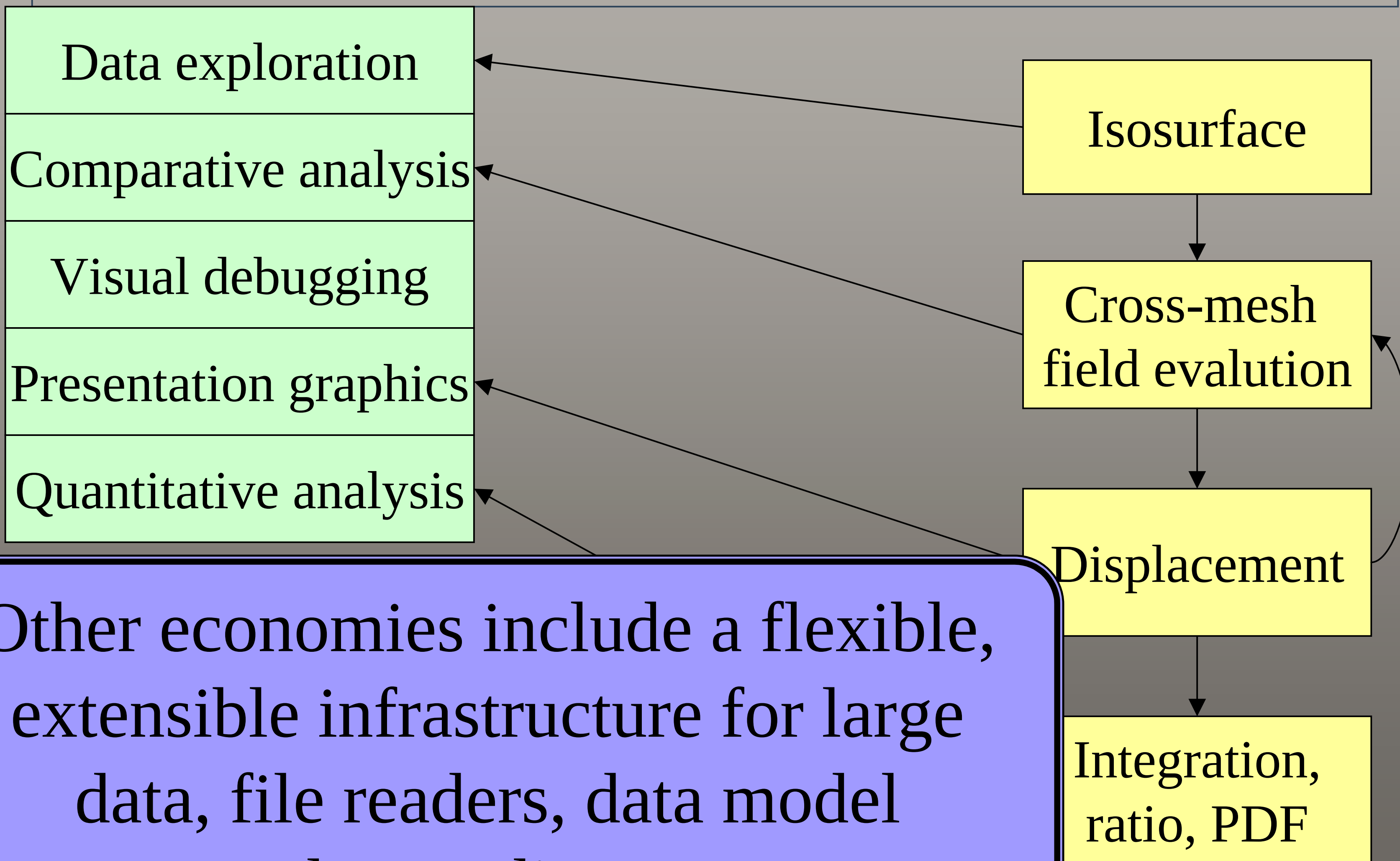
Integrate mass fraction over a wedge, then divide by surface area

Calculate joint PDF of ratios.
→ Produced 1D curve, done "real science"

# Each of these operations is turned into modules in an interoperable framework.



# The modules used span our use cases.




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.

| | Pure Parallelism | In Situ | Multires | Out-of-core |
|---|---|---|---|---|
| Data Exploration | ? | ? | ? | ? |
| Quantitative Analysis<br>Basic<br>Advanced<br>Physics-based | ? | ? | ? | ? |
| Comparative Analysis<br>Image-Based<br>Data-Level<br>Topological | ? | ? | ? | ? |
| Visual Debugging<br>Individual Queries<br>Aggregate Queries | ? | ? | ? | ? |
| Moviemaking<br>Presentation-oriented<br>Exploration-oriented | ? | ? | ? | ? |

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

■ = technique generally not viable for use case

| | Pure Parallelism | In Situ | Multires | Out-of-core |
|---|---|---|---|---|
| Data Exploration | Yes | No | Yes | No |
| Quantitative Analysis | | | | |
| Basic | Yes | Yes | No | Yes |
| Advanced | Yes | Sometimes | No | Yes |
| Physics-based | Yes | Sometimes | No | Yes |
| Comparative Analysis | | | | |
| Image-Based | Yes | No | No | Yes |
| Data-Level | Yes | No | No | Yes |
| Topological | Yes | No | No | Yes |
| Visual Debugging | | | | |
| Individual Queries | Yes | No | Yes | Yes |
| Aggregate Queries | Yes | Yes | No | No |
| Moviemaking | | | | |
| Presentation-oriented | Yes | Yes | No | Yes |
| Exploration-oriented | Yes | Yes | Yes | Yes |

# Not all processing techniques are equal

In situ

Multi-resolution

Acceptable

Cost prohibitive

Pure parallelism

Out-of-core

Computer-cost too high

Runtime-cost too high

= Primary option, = backup option

| | Pure Parallelism | In Situ | Multires | Out-of-core |
|---|---|---|---|---|
| Data Exploration | Yes | No | Yes | No |
| Quantitative Analysis | | | | |
| Basic | Yes | Yes | No | Yes |
| Advanced | Yes | Sometimes | No | Yes |
| Physics-based | Yes | Sometimes | No | Yes |
| Comparative Analysis | | | | |
| Image-Based | Yes | No | No | Yes |
| Data-Level | Yes | No | No | Yes |
| Topological | Yes | No | No | Yes |
| Visual Debugging | | | | |
| Individual Queries | Yes | No | Yes | Yes |
| Aggregate Queries | Yes | Yes | No | No |
| Moviemaking | | | | |
| Presentation-oriented | Yes | Yes | No | Yes |
| Exploration-oriented | Yes | Yes | Yes | Yes |

# 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)

■ = In VisIt, ■ = needed & not in VisIt

| | Pure Parallelism | In Situ | Multires | Out-of-core |
|---|---|---|---|---|
| Data Exploration | Yes | No | Yes | No |
| <u>Quantitative Analysis</u> | | | | |
| Basic | Yes | Yes | No | Yes |
| Advanced | Yes | Sometimes | No | Yes |
| Physics-based | Yes | Sometimes | No | Yes |
| <u>Comparative Analysis</u> | | | | |
| Image-Based | Yes | No | No | Yes |
| Data-Level | Yes | No | No | Yes |
| Topological | Yes | No | No | Yes |
| <u>Visual Debugging</u> | | | | |
| Individual Queries | Yes | No | Yes | Yes |
| Aggregate Queries | Yes | Yes | No | No |
| <u>Moviemaking</u> | | | | |
| Presentation-oriented | Yes | Yes | No | Yes |
| Exploration-oriented | Yes | Yes | Yes | Yes |

# 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.