Scalability and Power Efficiency of In-Situ Analysis Workflows

Aaditya G. Landge[1], Peer-Timo Bremer[2], and Valerio Pascucci[1]

Collaborators: Attila Gyulassy[1], Sidharth Kumar[1], Janine C. Bennet[3], Hemanth Kolla[3], Jacqueline Chen[3], Ivan Rodero[4], Manish Parashar[4]

[1] SCI Institute, University of Utah, [2] Lawrence Livermore National Laboratory, [3] Sandia National Laboratory, [4] Rutgers University

www.sci.utah.edu

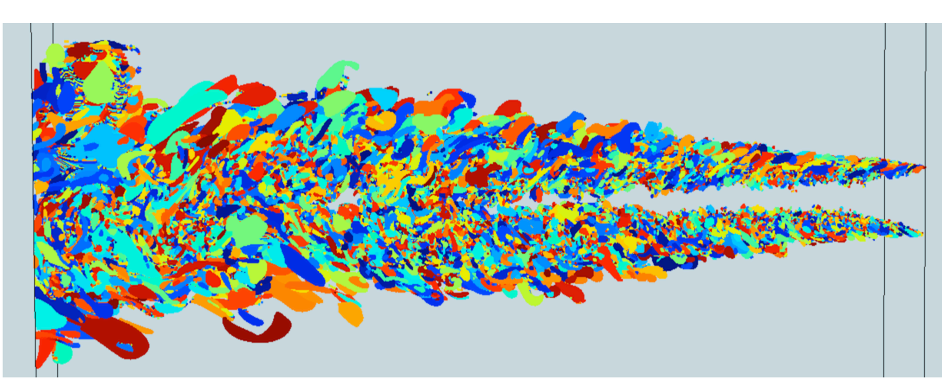
Abstract

In-situ analytics is becoming an integral part of the simulation pipeline. As only the features of interest are stored during in-situ analysis, features can be extracted at a much higher temporal frequency enabling deeper scientific insight.

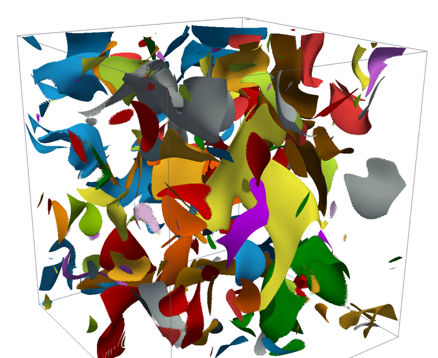
In this work, we analyze in-situ analysis workflows in terms of scalability and power efficiency that use the same compute resources as the on-going simulation. For this study, we make use of in-situ feature extraction using segmented merge tree[1] run at full machine scale on Titan.

Scientific Impact of In-Situ Analysis

Large scale in-situ feature extraction of S3D simulations :

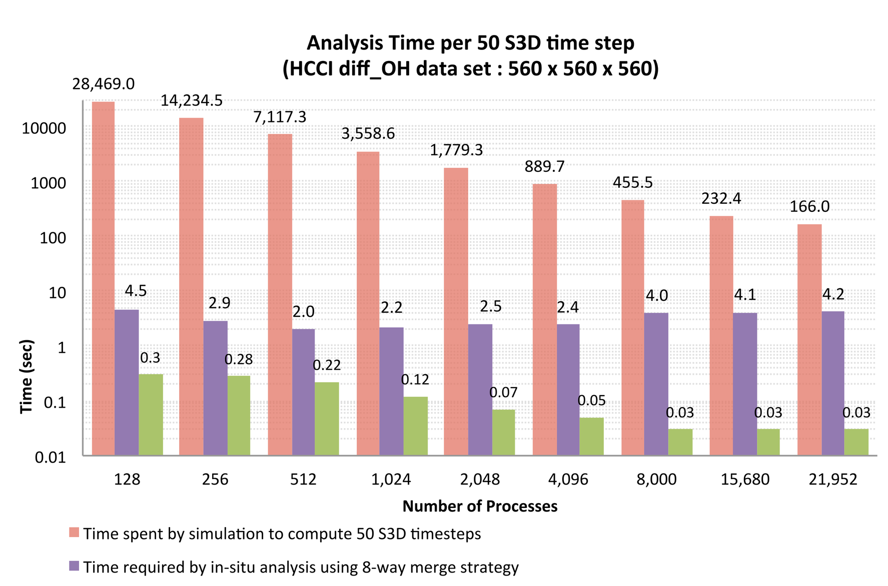


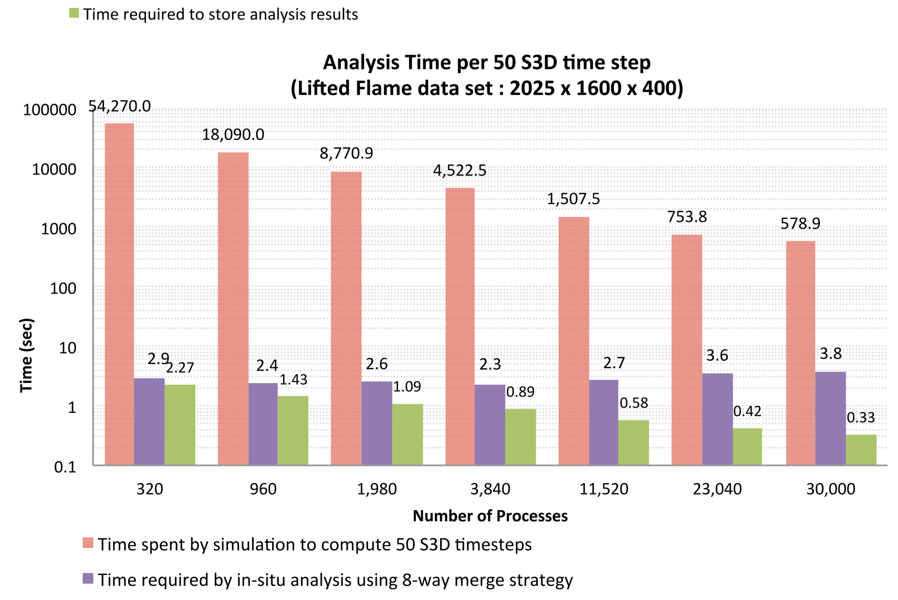
*Extinction regions from a **Lifted Ethylene Jet Flame** simulation used to investigate turbulent flames in direct injection stratified spark ignition engines for commercial boilers.*



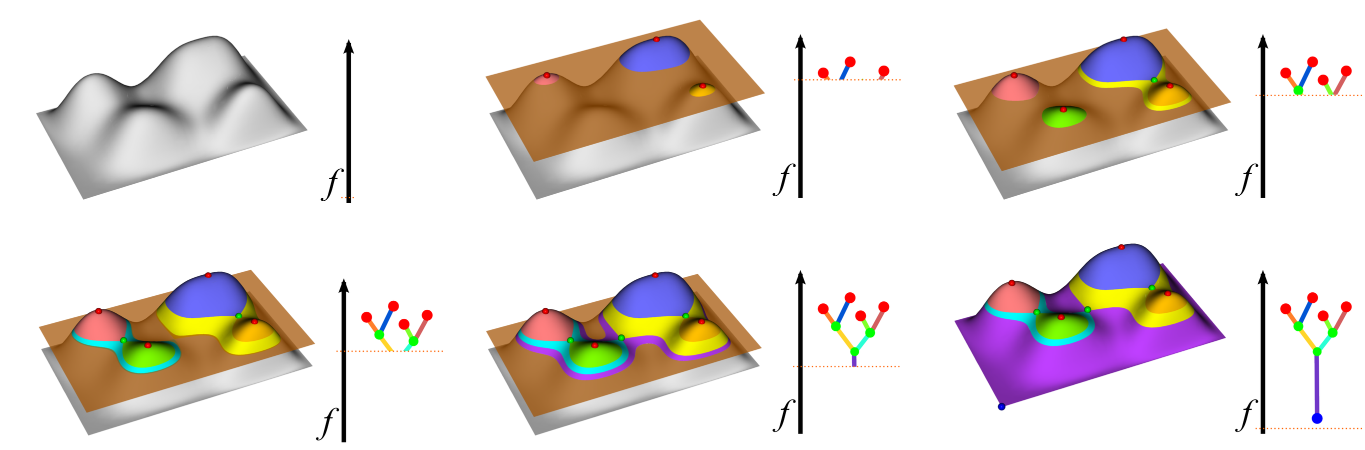
*Burning cells in **Homogeneous Charge Compression Ignition (HCCI)** in which a lean, premixed fuel-air mixture is compressed until it ignites spontaneously in many separate locations.*

- Feature extraction at every 50th time step instead of 500th time step – **10x increase in analysis frequency**
- **Takes less than 1% of the simulation time** for every 50 time steps
- Enables capture of intermittent and short lived features





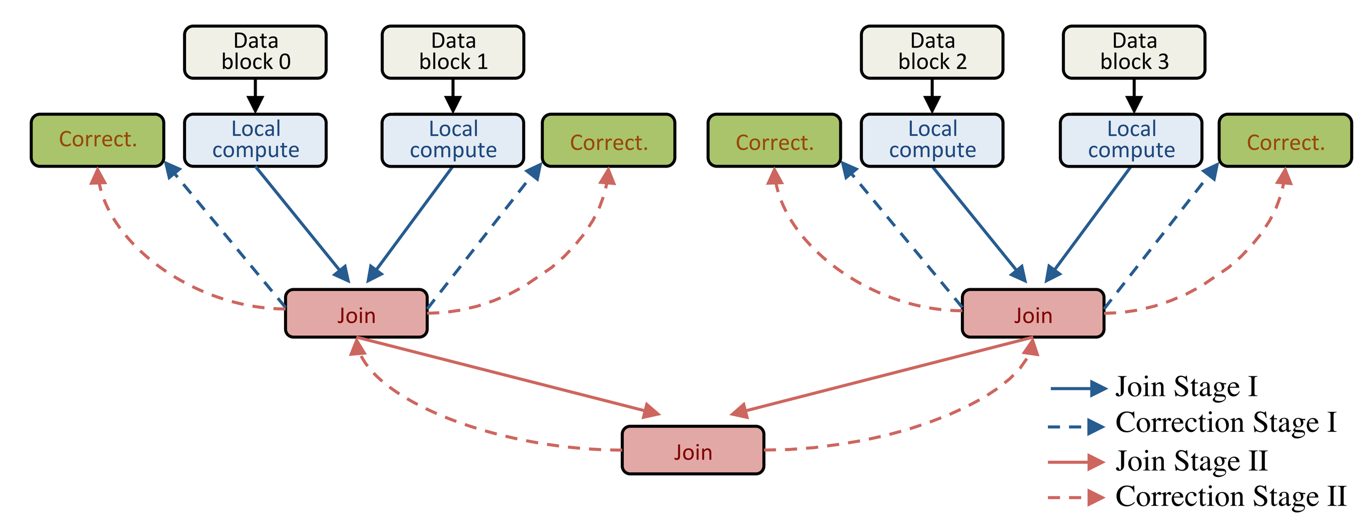
Segmented Merge Tree



A merge tree tracks the evolution of connected components of the domain as the function range is swept from ∞ to $-\infty$.

In-Situ Distributed Merge Tree Computation

The computation is composed of the following phases in k-way merge hierarchy:

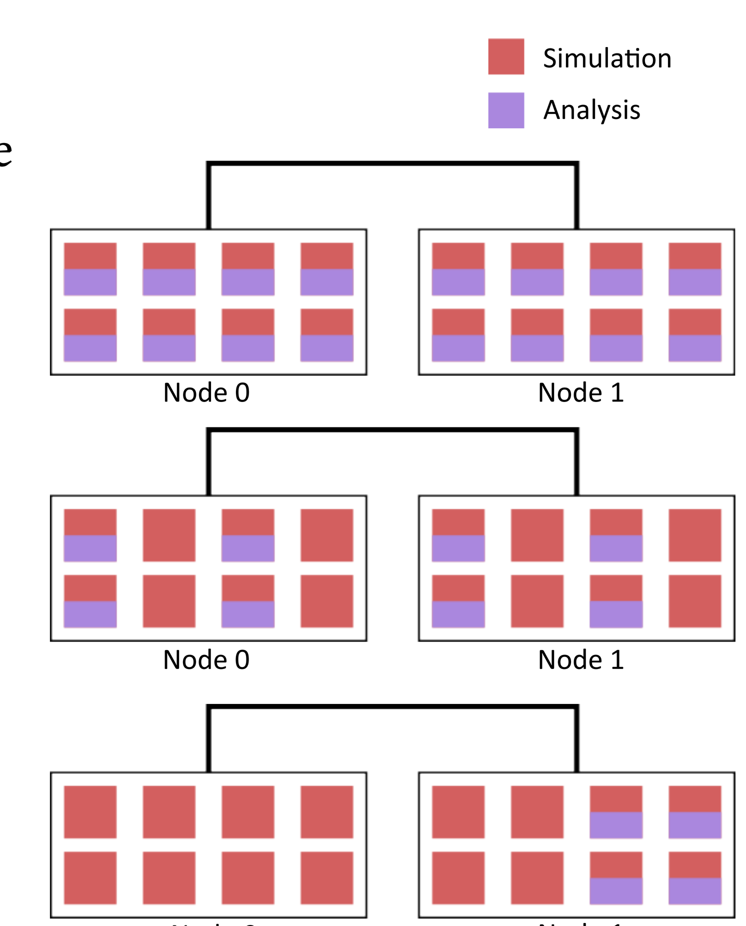


1. **Local Compute** - Compute a *local merge tree* for each decomposed data block
2. **Join** – Merge the shared boundaries of neighboring local trees
3. **Correction** – Correction of local trees based on joined boundary

In-Situ Analysis Workflows

Using the same compute resources as the simulation:

1. Using all the compute cores used by the simulation
2. Using fewer cores per node and on-node data movement
3. Using subset of the nodes and off-node data movement

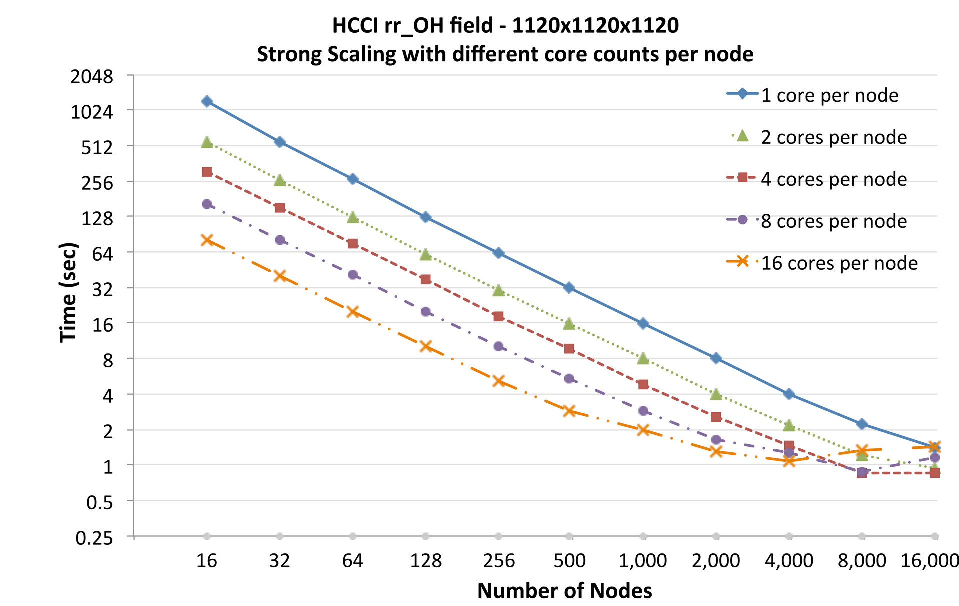


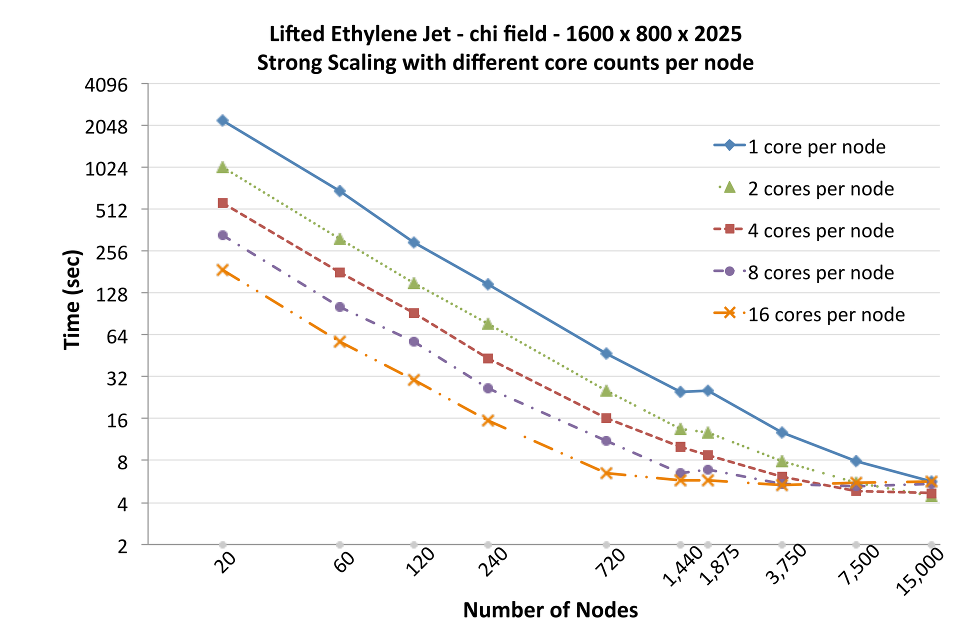
Performance and Scalability

Test System:

Titan Cray XK7 at ORNL

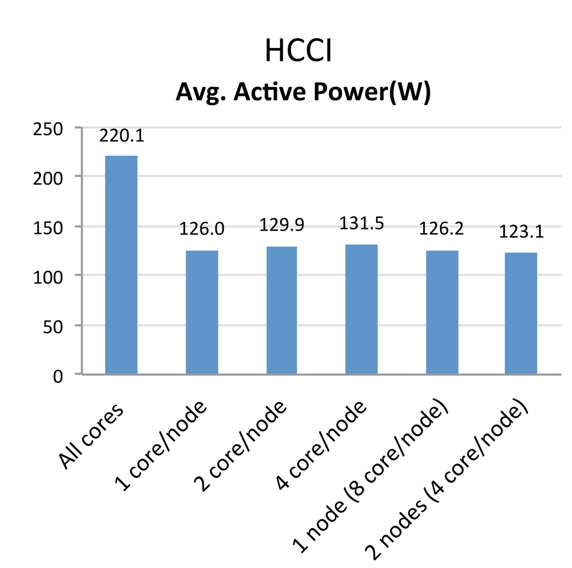
- Using all cores per node gives best performance at lower node counts
- At higher node counts, reduction in cores used per node gives better performance and scalability

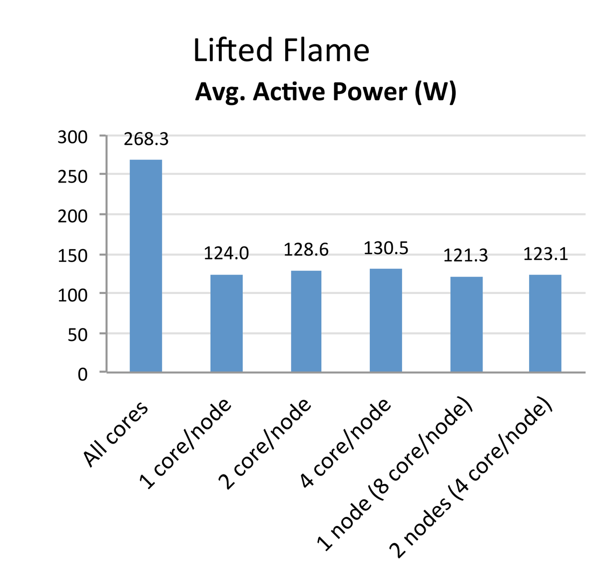




Power Analysis

Fine grained power measurements of appropriately downscaled problem using the CAPER cluster at Rutgers University





- Using **fewer cores per node and on-node data movement is the optimum in-situ analysis workflow** in terms of scalability as well as power efficiency

Future Work

- In-situ merge tree computation for Adaptive Mesh Refinement (AMR) meshes
- Resilient merge tree computation
- Quality vs. Resiliency trade-offs for in-situ analysis

References

[1] A. G. Landge, V. Pascucci, A. Gyulassy, J. C. Bennett, H. Kolla, J. Chen, and P.-T. Bremer. In-situ feature extraction of large scale combustion simulations using segmented merge trees. *In Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC '14*

This work was performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344 (LLNL-POST-).