Interactive Linked Visualizations for Performance Analysis of Heterogeneous Computing Clusters

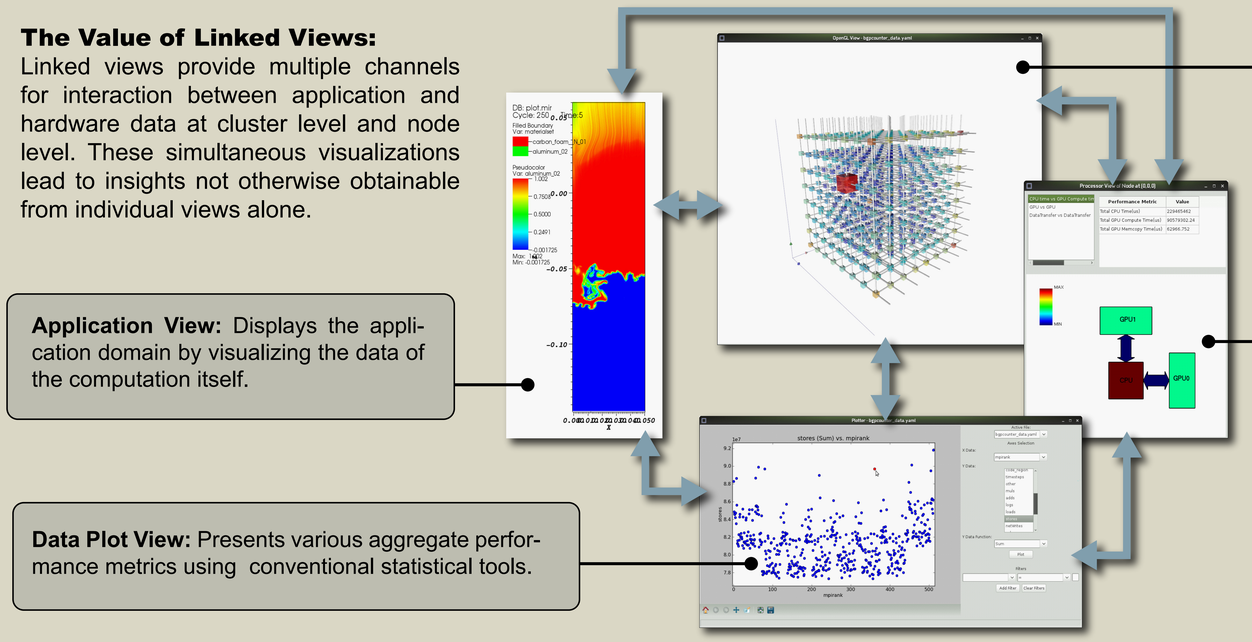
Aaditya Landge[1], Joshua A. Levine[1], Peer-Timo Bremer[2], Martin Schulz[2], Todd Gamblin[2], Abhinav Bhatele[2], Katherine E. Isaacs[3], Valerio Pascucci[1]
[1]Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT,
[2]Lawrence Livermore National Laboratory, Livermore, CA
[3]University of California, Davis

Abstract:
Performance data obtained from compute clusters is necessarily complicated because data is not just collected from a single node but from multiple interacting nodes, potentially with several cores each. Further, heterogeneous clusters, often in the form of nodes combining one ore more CPUs working together with several GPUs, are becoming more commonplace and are leading to even more complexity. These characteristics pose a serious challenge to the analysis and improvement of application performance.

We present a tool that assists performance analysis by visualizing performance data with the help of various linked views that:

- Draw correlations between the domain decomposition of the application and collected performance data;
- Enable views of the data at various granularities within the appropriate context;
- Provide a combined visualization of performance data from the CPUs as well as GPUs; and
- Increase the intuition behind the analysis of performance data.

The Value of Linked Views:
Linked views provide multiple channels for interaction between application and hardware data at cluster level and node level. These simultaneous visualizations lead to insights not otherwise obtainable from individual views alone.



Application View: Displays the application domain by visualizing the data of the computation itself.

Data Plot View: Presents various aggregate performance metrics using conventional statistical tools.

3D Cluster Level View: Represents the cluster of compute nodes spatially. Each cube is a compute node. Cubes are colored based on per node performance data.

Node Level View: Shows performance data for a single CPU, its GPUs, and the data transfer buses that connect them.

Case Study:
Performance behavior a of Hybrid CPU-GPU Solver for Gradient Domain Processing of Massive Images [2]

Application Details:
Seamless stitching of gigapixel panoramas through a parallel implementation using MPI and CUDA.

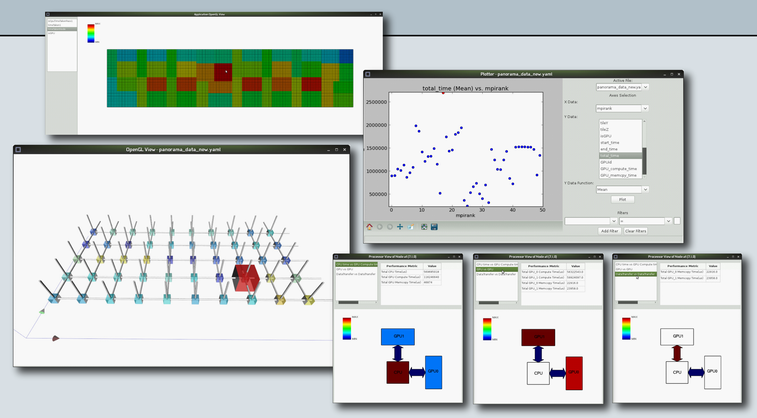
The algorithm decomposes the input images into 1024 X 1024 pixel sized tiles that are distributed across all MPI processes. Each of the MPI process then processes some tiles on a CPU or on the GPUs depending on available resources.

Application Domain:
A 126,826 X 29,633, 3.27 gigapixel panorama decomposed into
124 X 29 spatial tiles.

Hardware Domain: A 50 node cluster, each node comprised of an Intel Xeon X5550 2.6Ghz Processor having 8 cores and 2 NVIDIA Tesla T10 GPUs. This cluster is represented as a grid of 10 X 5 nodes.

Key questions answered:

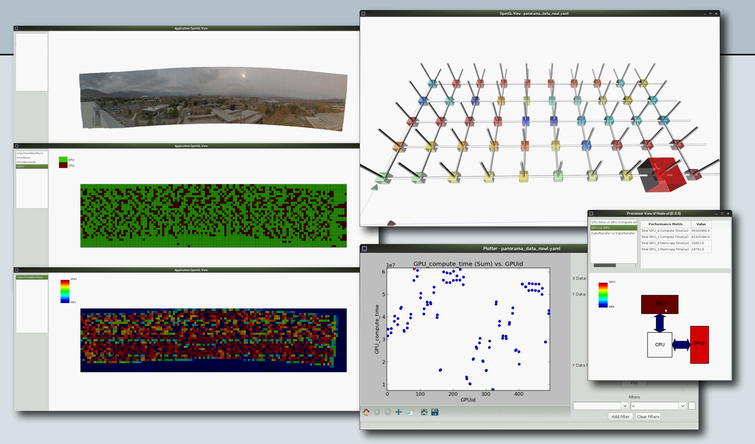
- How was the load balanced on the cluster?
- Which node took the maximum time?
- Was the reason for the longer duration because of some characteristic of the input?



Application View: Data partitioning on the cluster and time taken by each node. A click on the partition automatically highlights the node in the other views.
Cluster Level View: Each node colored by time taken. A selected node is highlighted with an enlarged red cube.
Data Plot View: Plot of total time of each node vs. the MPI rank. The data point for the selected node is highlighted in red.
Node Level Views of the selected node(from top to bottom): CPU time vs. Overall GPU compute time, GPU0 compute time vs. GPU1 compute time, Data Bus0 transfer time vs. Data Bus1 transfer time.

Key questions answered:

- How was the computation balanced between CPUs and GPUs?
- What was the time taken by each tile to be computed?
- Which GPU spent the maximum time doing computations?



Application View (top): Final image generated by the application.
Application View (middle): Image tiles, colored by if they were processed by CPUs or GPUs.
Application View (bottom): Time taken by the tiles processed on GPUs.
Cluster View: Each node colored by GPU compute time. Node with max GPU compute time is selected and highlighted with an enlarged red cube.
Data Plot View: Plot of GPU compute time vs. GPU ID.
Node Level View: Each GPU in the node colored by the GPU compute time.

References:
[1] M. Schulz, J. A. Levine, P-T Bremer, T. Gamblin, and V. Pascucci. **Interpreting performance data across intuitive domains.** International Conference Parallel Processing, Sept. 2011.
[2] S. Philip, B. Summa, P-T. Bremer, V. Pascucci. **Parallel Gradient Domain Processing of Massive Images.** Eurographics Symposium on Parallel Graphics and Visualization, Apr. 2011.

LLNL-POST-518831 - This work is performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.