Exploring Performance Data with Boxfish

Katherine E. Isaacs*, Aaditya G. Landge†, Todd Gamblin‡, Peer-Timo Bremer‡, Valerio Pascucci†, Bernd Hamann*

‡Center for Applied Scientific Computing, Lawrence Livermore National Laboratory, Livermore, CA 94551 USA
†Scientific Computing and Imaging Institute, The University of Utah, Salt Lake City, UT 84112 USA
*Institute for Data Analysis and Visualization, Department of Computer Science, University of California, Davis, CA 95616 USA
E-mail: keisaaacs@ucdavis.edu, {aaditya, pascucci}@sci.utah.edu, {tgamblin, ptbremer}@llnl.gov, hamann@cs.ucdavis.edu

The growth in size and complexity of scaling applications and the systems on which they run pose challenges in analyzing and improving their overall performance. With metrics coming from thousands or millions of processes, visualization techniques are necessary to make sense of the increasing amount of data. To aid the process of exploration and understanding, we announce the initial release of Boxfish, an extensible tool for manipulating and visualizing data pertaining to application behavior. Combining and visually presenting data and knowledge from multiple domains, such as the application's communication patterns and the hardware's network configuration and routing policies, can yield the insight necessary to discover the underlying causes of observed behavior. Boxfish allows users to query, filter and project data across these domains to create interactive, linked visualizations.

I. Projecting Data Across Domains

We describe the association of elements that exist in one domain with the elements of another as a *projection*. A map file which associates integer MPI ranks with coordinate-denoted hardware nodes and threads is an example of a commonly used projection. Schulz et al. [1] advocated the use of projections in interpreting performance data and defined three domains of interest – hardware, application and communication. The hardware domain includes performance counters. The application domain includes information relating to the application, such as physics measurements in a simulation or matrix properties in a linear algebra library. The communication domain includes messages sent among subsets of processors. Boxfish recognizes these domains by default, but contributed modules may add others.

Boxfish is designed to support the projection of data across domains. When filters or queries are written requiring attributes from multiple domains, or when a view requires attribute information in a native domain, Boxfish searches its available projections to make the necessary transformations. This allows users to view data such as the load on nodes which had a certain range of values in a previous run or the average wait time for communicators in a particular phase of the application. Data tables may have default preferred projections. Projections can be added from files, created based on data attributes, or composed from existing ones. More projections may be added through future or contributed modules.

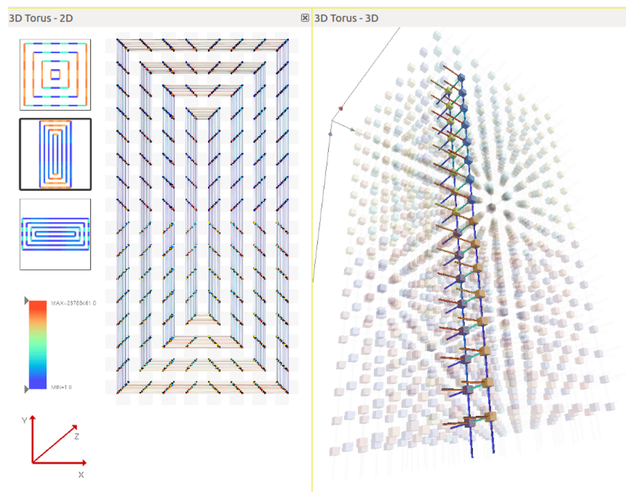

Fig. 1. A 3D torus network represented in 2D (left) and 3D (right). Both views represent elements of the hardware domain. However, nodes are colored by their sub-communicators, which belong to the communication domain. Links are colored by the number of packets sent over them. These views are rendered side by side in Boxfish, indicating they are siblings in the filter hierarchy and show the same data. In the 2D view, selected nodes are displayed at a slightly larger size. In the 3D view, the same nodes are selected and highlighted by their relative opacity.

Figure 1 shows a projection from the communication domain onto the hardware domain. The nodes of the hardware domain are colored by the values of the sub-communicators in the communication domain. This particular case is a one-to-many projection, where a single sub-communicator maps to many nodes. Boxfish also handles many-to-one projections with a choice of several functions to appropriately aggregate the results.

II. Filter Hierarchy in Boxfish

The data flow in Boxfish follows the hierarchical grouping and coordination of filters and views as implemented in Epinome [2], an application for exploring epidemiological scenarios. This system allows both the data manipulation and the view interactions to be applied simultaneously to several views at once. Users can move views and groups of views

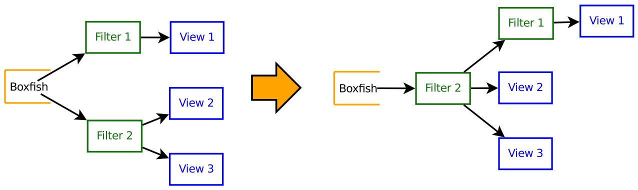

Fig. 2. Two potential Boxfish configurations. Left: the user has three views, two of which share a filter on their information. Right: the user has moved the Filter 1 subtree under Filter 2, so that View 1 receives its data after both filters have been applied.

anywhere in the hierarchy, dynamically changing the presented data, as illustrated in Figure 2.

When elements are selected in one view, corresponding elements in other views may be automatically highlighted depending on their position in the hierarchy and the policies of each subtree. Though the elements shown in each view may not be the same, selection in one can induce selection in the others if projections from the first view exist.

III. Boxfish Views

Boxfish is designed to facilitate the addition of new views by unifying the processing and manipulating of input data and handling shared view actions such as coloring, highlighting and rotating. Details on writing a Boxfish view are available in the user manual. We briefly describe the initial set of Boxfish views.

Torus/Mesh 3D – Three-dimensional torus and mesh networks are represented in their conventional configurations. In the case of the torus, wrap-around links are shown from one of their two end-nodes. The location of this 'seam' in any of the dimensions can be selected arbitrarily. Attributes are displayed on nodes and links by color. Figure 1 (right) shows this view.

Torus/Mesh 2D – Three-dimensional torus and mesh networks are arranged on a 2D plane to eliminate occlusion and ease selection. To achieve this representation, some of the links are minimized or omitted. Users can change which dimension is minimized and which sets of links are omitted. All three configurations are shown in overview 'minimaps' that aggregate attribute values on the links. Like its 3D counterpart, attributes are shown in this view by coloring the nodes and links. This view is shown in Figure 1 (left).

Plotter – Boxfish's default plotting module can create scatter plots and histograms of (potentially aggregated) data attributes. Attributes from one or multiple domains may be plotted against the elements or attributes of any domain for which a projection exists. The plots offer a familiar method of looking at the data and selecting features of interest which can then be automatically highlighted in other views.

IV. Boxfish in Practice

Bhatele et al. [3] used Boxfish's 3D torus view to help identify the cause of a scaling problem in SAMRAI [4], [5], an adaptive mesh refinement (AMR) library. Nodes that spent the most time in a load balancing phase appeared clustered in the 3D torus view, providing insight into the relationship between the slower processes and overlay communication network being used.

Boxfish's 2D torus view has been used to better understand network behavior [6], [7] in pF3D [8], [9], a multi-physics laser-plasma interaction simulation. The view showed the differences in traffic load in the various torus directions given various node mappings. The 3D torus view was also used [7] to verify the topological layout of sub-communicators under particular node mappings.

V. Conclusion

Intelligent visualization of performance data can yield insights necessary to optimize applications. Utilizing projections across the multiple domains which affect overall performance enables the discovery of interactions between contexts. Boxfish manages these projections and visualizations for creative, convenient exploration of performance data.

Acknowledgments

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-CONF-568312. This research is supported in part by the Department of Energy Office of Science Graduate Fellowship Program (DOE SCGF), administered by ORISE-ORAU under contract no. DE-AC05-06OR23100.

References

[1] M. Schulz, J. Levine, P.-T. Bremer, T. Gamblin, and V. Pascucci, "Interpreting performance data across intuitive domains," in *Parallel Processing (ICPP), 2011 International Conference on*, September 2011, pp. 206–215.

[2] Y. Livnat, T.-M. Rhyne, and M. H. Samore, "Epinome: A visual-analytics workbench for epidemiology data," *IEEE Computer Graphics and Applications*, vol. 32, no. 2, pp. 89 – 95, 2012.

[3] A. Bhatele, T. Gamblin, S. H. Langer, P.-T. Bremer, E. W. Draeger, B. Hamann, K. E. Isaacs, A. G. Landge, J. A. Levine, V. Pascucci, M. Schulz, and C. H. Still, "Mapping applications with collectives over sub-communicators on torus networks," in *Proceedings of the 2012 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. IEEE Press, November 2012.

[4] R. D. Hornung and S. R. Kohn, "Managing application complexity in the samrai object-oriented framework," *Concurrency and Computation: Practice and Experience*, vol. 14, no. 5, pp. 347–368, 2002.

[5] B. T. Gunney, A. M. Wissink, and D. A. Hysom, "Parallel clustering algorithms for structured amr," *Journal of Parallel and Distributed Computing*, vol. 66, no. 11, pp. 1419 – 1430, 2006.

[6] A. Bhatele, T. Gamblin, K. E. Isaacs, B. T. N. Gunney, M. Schulz, P.-T. Bremer, and B. Hamann, "Novel views of performance data to analyze large-scale adaptive applications," in *Proceedings of the 2012 ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*, ser. SC '12. IEEE Press, November 2012.

[7] A. G. Landge, J. A. Levine, K. E. Isaacs, A. Bhatele, T. Gamblin, M. Schulz, S. H. Langer, P.-T. Bremer, and V. Pascucci, "Visualizing network traffic to understand the performance of massively parallel simulations," *IEEE Transactions on Visualization and Computer Graphics (InfoVis 2012)*, vol. 18, no. 12, pp. 2467–2476, December 2012.

[8] C. H. Still, R. L. Berger, A. B. Langdon, D. E. Hinkel, L. J. Suter, and E. A. Williams, "Filamentation and forward brillouin scatter of entire smoothed and aberrated laser beams," *Physics of Plasmas*, vol. 7, no. 5, p. 2023, 2000.

[9] R. L. Berger, B. F. Lasinski, A. B. Langdon, T. B. Kaiser, B. B. Afeyan, B. I. Cohen, C. H. Still, and E. A. Williams, "Influence of spatial and temporal laser beam smoothing on stimulated brillouin scattering in filamentary laser light," *Phys. Rev. Lett.*, vol. 75, no. 6, pp. 1078–1081, Aug 1995.