Report from Dagstuhl Perspectives Workshop 14022

Connecting Performance Analysis and Visualization to Advance Extreme Scale Computing

Edited by

Peer-Timo Bremer[1,3], Bernd Mohr[2], Valerio Pascucci[3], and Martin Schulz[4]

1 **Lawrence Livermore National Laboratory*, US,** bremer5@llnl.gov
2 **Jülich Supercomputing Centre, DE,** B.Mohr@fz-juelich.de
3 **University of Utah – Salt Lake City, US,** pascucci@sci.utah.edu
4 **Lawrence Livermore National Laboratory*, US,** schulz6@llnl.gov

Abstract

In the first week of January 2014 Dagstuhl hosted a Perspectives Workshop on "Connecting Performance Analysis and Visualization to Advance Extreme Scale Computing". The event brought together two previously separate communities – from Visualization and HPC Performance Analysis – to discuss a long term joined research agenda. The goal was to identify and address the challenges in using visual representations to understand and optimize the performance of extreme-scale applications running on today's most powerful computing systems like climate modeling, combustion, material science or astro-physics simulations.

Perspectives Workshop January 5–10, 2014 – http://www.dagstuhl.de/14022
1998 ACM Subject Classification C.4 Performance of Systems, D.1.3 Concurrent Programming (Parallel programming), D.4.8 Performance, I.3 Computer Graphics
Keywords and phrases Large scale data presentation and analysis, Exascale class machine optimization, Performance data analysis and root cause detection, High dimensional data representation
Digital Object Identifier 10.4230/DagRep.4.1.17

1 Executive Summary

Peer-Timo Bremer
Bernd Mohr
Valerio Pascucci
Martin Schulz

Over the last decades an incredible amount of resources has been devoted to building ever more powerful supercomputers. However, exploiting the full capabilities of these machines is becoming exponentially more difficult with each new generation of hardware. In the systems coming online at this moment, application developers must deal with millions of cores, complex memory hierarchies, heterogeneous system architectures, high-dimensional network topologies as well as a host of other hardware details that may affect the performance of a

* Part of 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-TR-653252).

code. To help understand and optimize the behavior of massively parallel simulations a new subfield of computer science has grown devoted to developing tools and techniques to collect and analyze performance relevant data, such as execution time, operation counts, and memory or network traffic to help application developers pinpoint and ultimately fix performance problems. There now exist a number of standardized tools and APIs to collect a wide range of performance data at the largest scale. However, this success has created a new challenge, as the resulting data is far too large and too complex to be analyzed in a straightforward manner. While there exist some tools for performance analysis and visualization, these are predominately restricted to simple plots of the raw data and rely virtually exclusively on the users to infer connections between measurements and the observed behavior and to draw conclusions. Unfortunately, as the number of cores increases, this approach does not scale. The raw data is typically rather abstract, low-level, and unintuitive and it is difficult to understand within the context of the highly complex interaction of an application with the middle- and system software and the underlying hardware. For this reason, new automatic and more scalable analysis approaches must be developed to allow application developers to intuitively understand the multiple, interdependent effects that their algorithmic choices have on the resulting performance.

Following classical visualization mantra, the natural first step towards automatic analysis is to display an overview of the collected data to provide some insight into general trends. This helps both application developers and performance experts to form new hypotheses on potential causes of and solutions to performance problems. Furthermore, intuitive visualizations are highly effective in conveying the results of any analysis and thus are a valuable tool throughout the entire process. Unfortunately, visualizing performance data has proven challenging as the information is highly abstract, non-spatial, and often categorical. While some early attempts at including more advanced visualizations in performance tools have been proposed, these are rudimentary at best and have not found widespread adoption.

At the same time there exists a vibrant community in the area of information visualization and lately visual analytics that is exclusively aimed at developing techniques to visualize, illustrate, and analyze complex, non-spatial data. In particular, there exists a large body of work on general design principles of visualization tools, color spaces, and user interfaces as well as a wide array of common techniques that tackle a broad range of applications. The Dagstuhl Perspectives Workshop, for the first time, gathered leading experts from both the fields of visualization and performance analysis for joint discussions on existing solutions, open problems, and the potential opportunities for future collaborations.

The week started with a number of keynote sessions from well-known authorities in each area to introduce the necessary background and form a common baseline for later discussions. It soon became apparent that there exists a significant overlap in the common tasks and challenges in performance analysis and the abstract problem definitions and concepts common in visualization research. Subsequently, the workshop continued with short talks focusing on various more specific aspects of either existing challenges or potential solutions interspersed with increasingly longer group discussions. Theses extensive, inclusive, and in-depth exchanges ultimately shaped the second half of the workshop and in this form were only made possible through Dagstuhl's unique collaborative and discussion stimulating environment.

Ultimately, the workshop has started a number of collaborations and research projects between previously disparate fields with the potential of significant impact in both areas. Furthermore, the participants distilled the open challenges into three high-level recommendations: First, joined funding for the various open research questions. Second, support to build and foster a new community on the border of visualization and performance analysis. And Third, the need to better integrate the anticipated results into the entire lifecycle of a massively parallel application from design to optimization and production.

2 Table of Contents

3 Keynotes and Topic Introduction

3.1 Really Quick Introduction to Parallel Performance Analysis

Bernd Mohr (Jülich Supercomputing Centre, DE)

This presentation provides a really quick introduction to parallel performance analysis. First, the common parallel programming paradigms MPI (Message Passing Interface) and OpenMP (Open specification for Multi-Processing) are introduced. Then the basic terminology and methodology for parallel program instrumentation, measurement, analysis and visualization is introduced. Finally, the two main sorts of performance data, summary profiles and detailed event trace, and their typical visualizations are explained in more detail.

3.2 Information Visualization and Visual Analytics

Daniel A. Keim (Universität Konstanz, DE)

There is a wide range of information visualization techniques that may be useful in analyzing the performance of HPC applications. The talk provides a brief overview of interesting visualization techniques and elaborates on the role of analysis and visualization. In putting visual analysis to work on big data, it is not obvious what can be done by automated analysis and what should be done by interactive visual methods. In dealing with massive data, the use of automated methods is mandatory – and for some problems it may be sufficient to only usefully automated analysis methods, but there is also a wide range of HPC problems where the use of interactive visual methods is necessary. The talk demonstrates a number of successful visual analytics applications from IP network monitoring, financial analysis, time series and high-dimensional data analysis.

3.3 Visualizing Performance Profiles with CUBE

Felix Wolf (German Research School for Simulation Sciences, DE)

Joint work of Wolf, Felix; Abraham, Erika; Bhatia, Nikhil; Böhme, David; Dennis, John; Geimer, Markus; Hatem, Youssef; Lücke, Monika; Pulatova, Farzona; Saviankou, Pavel; Schumacher, Peter; Song, Fengguang; Szebenyi, Zoltán; Visser, Anke; Voigtländer, Felix; Wylie, Brian
URL http://www.scalasca.org/software/cube-4.x/download.html

In this talk, we presented CUBE, an interactive browser that supports the visual exploration of performance profiles from parallel applications. CUBE's design emphasizes simplicity by combining a small number of orthogonal features with a limited set of user actions. The talk first described the underlying performance data model and the general usage model. It then focused on the visualization of process topologies, including Cartesian meshes with three and more dimensions and with potentially irregular subdomains. As further visualization targets, the talk covered time-dependent performance behavior and irregular domains. Finally, the talk discussed how CUBE interoperates with other visualization tools such as Vampir.

3.4 Vampir: Process Visualization of High Performance Software

Holger Brunst (TU Dresden, DE)

Joint work of Brunst, Holger; Knüpfer, Andreas
Main reference H. Brunst, A. Knüpfer, "Vampir," in David A. Padua (ed.), Encyclopedia of Parallel Computing, pp. 2125–2129, Springer, 2011.
URL http://dx.doi.org/10.1007/978-0-387-09766-4_60

Performance optimization is a key issue for the development of efficient parallel software applications. Vampir is a framework for performance analysis, which enables developers to quickly study program behavior at a fine level of detail. Performance data obtained from a parallel program run can be analyzed with a collection of specialized performance views. Intuitive navigation and zooming are the key features of the tool, which help to quickly identify inefficient or faulty parts of a program code.An important and unique feature of Vampir is its intuitive and interactive graphical representation of detailed performance event recordings over time (timelines) and as aggregated profiles. Extensive searching and filtering capabilities allow to quickly identify critical bottlenecks. In contrast to traditional profiling the details that caused a problem remain close at hand. The performance charts include rich sets of performance information and can be customized to the needs of both beginners and experts.New high performance computing systems are currently designed with a constantly increasing number of processing entities (cores), which is motivated by Moore's law and physical limitations. Performance tool visualizations have to follow this trend in order to be helpful in the future. Some of the resulting scalability issues in performance data visualization are presented in this talk in the scope of the Vampir framework.

3.5 Gathering and Interpreting Performance Data Visualizations – Initial Scaffolding

Martin Schulz (Lawrence Livermore National Laboratory, US)

Performance data can be gathered from a variety of different sources. This talk provides a straw-man for classifying these into four domains: hardware, tasks, code, and application data. Visualizing this data can in some cases be done within the same domain the data was collected in. An example is the visualization of hardware properties such as network links in the hardware domain. In other cases, though, the data is more helpful after it has been projected into a different domain and visualized there. An example is performance data of a CFD code mapped into the application domain to understand performance artifacts caused by the physics being simulated. The talk concludes with a proposal for such mappings and open research questions in establishing such a domain/mapping model, which decouples the data collection and visualization domains.

3.6 A(n Opinionated) Tour of InfoVis

Carlos E. Scheidegger (AT&T Labs Research – New York, US)

In this talk I present the following polemic: the entirety of information visualization design can be summarized as identifying mathematical structures in the input data space, and making sure they match in the visual encoding. In this sense, effective data visualizations are effective not because our eyes have more bandwidth, but rather because this visual representation of the data is actually closer its true nature. In addition, formulating our visualizations explicitly to elicit different structures in the data naturally leads to a classification of a large portion of dimensionality reduction techniques and graph drawing. Crucially, we don't require a rigid taxonomy to achieve this classification, so new data modalities can be comfortably incorporated by thinking about the properties of the input space that can or should be preserved.

3.7 Visual Analytics for High-Dimensional Data

Klaus Mueller (Stony Brook University and SUNY Korea, US)

Extreme scale computing can generate vast amounts of profile and trace data with many variables. In addition, the underlying machine has processors that are interconnected in a multi-dimensional (3-5) torus topology. The challenge is to visualize these data in such a way that multivariate relationships can be revealed, and optionally in the context of the multi-dimensional topology of the inter-connection network. I presented the underlying principles of visual analytics which is formed by a synergistic triad of visualization, data analytics, and interaction. The latter puts the analysts directly into the loop of the analytics data exploration process. I discussed various high-dimensional visualization techniques that come to live by the element of user interaction. All of these have data analytics and even machine learning algorithms under the hood to assist the analyst in the online reasoning process [5, 6]. Specifically, I discussed the method of (1) parallel coordinates as a ways to visualize the raw data along parallel dimension axes [1, 2], (2) correlation maps that allow users to see dependencies of patterns in the data [3], and (3) dynamic scatterplots that exploit motion parallax for an interactive high-dimensional viewing experience [4]. I also discussed our recent software framework, called "The ND-Scope" [8], which incorporates various facilities for high-dimensional data exploration and reasoning with high-dimensional data, also in the context of the earth's geography [7].

References

1 A. Inselberg, B. Dimsdale. *Parallel Coordinates: A Tool for Visualizing Multi-dimensional Geometry*. IEEE Visualization, pp. 361–378, 1990.
2 K. McDonnell and K. Mueller. *Illustrative Parallel Coordinates*. Computer Graphics Forum, 27(3):1031–1038, 2008.
3 Z. Zhang, K. McDonnell, K. Mueller. *A Network-Based Interface for the Exploration of High-Dimensional Data Spaces*. IEEE Pacific Vis, Songdo, Korea, pp. 17–24, March, 2012.

4 J. Nam, K. Mueller. *TripAdvisorN-D: A Tourism-Inspired High-Dimensional Space Exploration Framework with Overview and Detail*. EEE Transactions on Visualization and Computer Graphics, 19(2): 291–305, 2013.
5 S. Garg, IV Ramakrishnan, K. Mueller. *A Visual Analytics Approach to Model Learning*. IEEE Conference on Visual Analytics Science and Technology (VAST'10), pp. 67–74, Salt Lake City, October, 2010.
6 S. Garg, E. Nam, IV. Ramakrishnan, K. Mueller. *Model-Driven Visual Analytics*. IEEE Symposium on Visual Analytics Science and Technology (VAST'08), pp. 19–26, 2008.
7 Z. Zhang, X. Tong, K. McDonnell, A. Zelenyuk, D. Imre, K. Mueller. *An Interactive Visual Analytics Framework for Multi-Field Data in a Geo-Spatial Context*. Tsinghua Science and Technology on Visualization and Computer Graphics, 18(2), April, 2013.
8 http://nd-scope.net/

3.8 Information Visualization for Performance Analysis – Initial Scaffolding

Peer-Timo Bremer (Lawrence Livermore National Laboratory, US)

Building on previous talks on the foundations of performance analysis this talk will provide an initial grouping and classification of existing visualization and analysis approaches. The goal is to provide the audience an overview of general approaches in visualization that may be applicable to performance analysis. In particular, with respect to the definition of different data domains we will highlight existing approaches in various domains and group them by different visualization paradigms such as hierarchical / focus+ context views or the type of interactions they allow. Finally, we provide some examples of gaps in the current landscape of techniques to start the discussion on how these may ultimately be addressed.

3.9 Some Remarks on the "User's" Perspective

Hans-Joachim Bungartz (TU München, DE)

The performance analysis community develops tools, and the workshop aims at identifying advanced visualization techniques, esp. related to information visualization and to visual analytics, that can enhance performance analysis tools – in terms of functionality, but in particular in terms of usability for those who are expected to use performance analysis. So what is the perspective of these "users" on risks and potentials, on automation and interaction, and on complexity and possible gains with respect to the ease of working with performance analysis tools? This perspective is the mandate of that presentation.The talk takes this point of view by clarifying the background, research practice, interest, and behavioral patterns of different "user specimens"; by clarifying the notion of performance from an algorithmic point of view; by pointing out the extreme importance of (numerical) algorithms, concerning both discretization and solution, as a link between "code" and "application" that must not be forgotten; by showing some examples of how code development,evaluation, and optimization in a CSE/HPC context works, when performance analysis enters the stage typically, and what is out of reach for performance analysis (currently); by briefly depicting typical users' problems with performance analysis tools, their output data, and their current visualization capabilities.The talk ends with several somewhat provocative questions and statements, also taking into account the overview-type presentations on performance analysis and visualization from the workshop's first day.

4 Participant Contributions

4.1 Looking at Time-stamped Data

Judit Gimenez (Barcelona Supercomputing Center, ES)

Humans are visual creatures. Our brain is an expert on correlating, matching and extracting insight from visual information. In order to benefit from this capability the BSC Performance Tools (www.bsc.es/performance_tools) target to understand the applications behavior trough the analysis of detailed performance views. Our tools are based on traces to expose to the performance analyst all the details on the variability and distribution of the measured metrics. Paraver has only two types of views but a lot of flexibility to define and correlate them. Timelines provide the evolution along time and tables (histograms,profiles) a measurement of the metrics distribution. With Paraver, the analyst drives the full process having the power to decide how to explore the performance data generating and validating his/her hypothesis about the application behavior. The initial steps would be similar for all the analysis and are provided as basic methodology, the results of these steps would allow to decide the path to follow. To facilitate extracting insight from detailed performance data, during the last years we have been exploring the potential of applying data analytics techniques. Clustering, tracking and folding allow the performance analyst to identify the program structure, study its evolution and look at the internal structure of the computation phases. Paraver is a very flexible tool but it can also provide a very quick analysis of the application efficiency. A Paraver demo was given showing two complementary strengths of the tool: to demonstrate how using very few views and mouse clicks you can get a short diagnosis of your code as well as the ability of the tool to identify details and navigate through the collected data.

4.2 Extracting Logical Structure from MPI Traces

Todd Gamblin (Lawrence Livermore National Laboratory, US)

Joint work of Isaacs, Katherine; Gamblin, Todd; Bremer, Timo; Schulz, Martin; Bhatele, Abhinav; Hamann, Bernd
Main reference K. E. Isaacs, T. Gamblin, A. Bhatele, P.-T. Bremer, M. Schulz, B. Hamann, "Extracting logical structure and identifying stragglers in parallel execution traces," in Proc. of the 19th ACM SIGPLAN Symp. on Principles and Practices of Parallel Programming (PPoPP'14), pp. 397–398, ACM, 2014.
URL http://dx.doi.org/10.1145/2555243.2555288

Event traces are a valuable tool for understanding the performance of parallel programs, but analyzing large traces is difficult. Most current visualization techniques adopt a Gantt-chart like paradigm to show events over time. Unfortunately, laying out computation and communication events in the order of their wall clock time introduces a large amount of clutter, which obscures important information in the visualization. In this paper, we describe a method to extract logical structure from a trace, which maintains order in terms of Lamport happened-before relations. We use this structure to define a notion of lateness among peers in a logical step. We show that laying out processes in terms of logical time and coloring them by lateness removes clutter, highlights parallel dependency chains, and uncovers the sources of parallel delays.

4.3 Prospective Visual Analysis Tools for Pattern Detection in Computing Performance Data

Tobias Schreck (Universität Konstanz, DE)

The analysis of computing performance data is an important problem to help understand computing resource consumption and to improve computing processes. It is also a difficult problem due to the size and complexity of data arising.Measurements may include e.g., large amounts of time- or network-oriented data.The visualization of the raw measurement data may not be most effective for the analysis, but techniques based on data aggregation and interactive filtering maybe needed. We present a selection of approaches from various application domains, in which data reduction, data visualization and interactive search are combined, and which may be a starting point for developing computing performance data analysis tools. Specifically, we survey techniques for exploration of clusters of network graphs and time-dependent scatter plot charts; for clustering of large numbers of heat map displays; and example-based search inline chart and scatter plot data. The aim of this talk is to survey a number of tools as to stimulate discussion on their application and extension for the analysis of computing performance data.

4.4 Mathematical Modeling of HPC Performance Analysis

Hans Hagen (TU Kaiserslautern, DE)

Analysis and simulation of complex physical phenomena, for example by CFD calculations, involves high performance computing on massively parallel machines. Often data is divided into subregions, transferred to a cluster or supercomputer and on to computing nodes. Overall performance of the computation is often measured in terms of efficiency, which is defined by the ratio of time used to actually solve the scientific problem by total run time. Performance heavily depend on various aspects like, for example, software implementation, system set-up, system resource occupancy during run time, communication behavior, and algorithmic parameter settings. In order to analyse HPC performance, various performance data, such as traces or energy use, are logged during run time. Traces capture the course of events during an HPC session and provide "insight" on what is happening where and under which condition.

Naturally, there are many challenges to the visualization of HPC performance data. For example, the mere data size brings numerous algorithmic and technical challenges. Nevertheless, at a first glance, this is looks like a typical visual analytics problem. But larger problems emerge at second glance. What is the mathematical characterization of the involved objects? How is the space or topology defined? In this talk, we formulate open problems from a visualization perspective that can serve as a basis for an in-depth discussion of this community.

4.5 A Potential Approach to Address Scalability Problems of Classical Performance Analysis by Combining Techniques from Visualization and Performance Analysis

Matthias S. Mueller (RWTH Aachen, DE)

Joint work of Mueller, Matthias S.; Henschel, Bernd

The rapid advancement of HPC has lead to huge scalability requirements for performance analysis tools. Huge progress has been made in the last 15 years (my personal time period with close contact to visualization and performance tools).However, facing limited financial resources all tools had to make their individual compromise between ease of use and scalability. As a consequence all tools share some problems: (a) they tend to solve problems of the past and fail at the highest scale or latest architecture/software and (b) the required abstraction level to cope with the scalability is difficult to understand even for advanced users.After all, I believe that this is an inherent problem that cannot be solved. But I also believe that an approach that combines different technologies and allows the users to switch between different abstraction levels and perspectives/views will be of great value at extreme scale. I will show three examples where current tools lack the flexibility to support that and how techniques from visualization can help to achieve this:
- Interactivity (Brushing , ...) and linked displays/views [2]
- Uncertainties
- Graphs

Especially the linked display/views technology combined with brushing us a key technology to maintain the understanding of the data that is analyzed and achieve the necessary insight. Because after all, performance analysis is not about tiny performance improvements, but it wants to achieve a general understanding of complex applications with their algorithms on complicated hardware architectures.

References

1 B. Shneidermann *The eyes have it: a task by datatype taxonomy for information visualizations.* Proceedings of IEEE Symposium onVisual Languages, 1996.
2 Laramee, Robert S., et al. *Visual Analysis and Exploration of Fluid Flow in a Cooling Jacket.* Visualization, 2005. VIS 05. IEEE. IEEE, 2005.2005.

4.6 Scalable Representations for Performance Data

Christopher Muelder (University of California – Davis, US)

Joint work of Muelder, Christopher; Sigovan, Carmen; Ma, Kwan-Liu; Ross, Robert; Gygi, Francois; Cope, Jason; Lang, Sam; Iskra, Kamil; Beckman, Pete

Many performance visualization approaches rely on representations of per-node or per-process level data over time to instantiate an initial view of the data, but such representations generally can only depict minimally small examples, and rarely scale to the size of modern machines. We have been developing a number of representations of various parallel performance data classes aimed at being capable of scaling up to handle current and forthcoming scales, such as rank agnostic views, animated views, or system level metric behavioral similarity views. While these representations may not be as intuitive as classical representations, they have enabled views of data at scales unheard of before. Integration of these and similar techniques with more complex analytics and more intuitive detailed representations would lead to much more scalable visual analytic approaches for handling and exploring large scale performance data.

References

1 Carmen Sigovan, Chris Muelder, and Kwan-Liu Ma *Visualizing Large-scale Parallel Communication Traces Using a Particle Animation Technique.* In Computer Graphics Forum 32(3), 2013, pages 141–150 (EuroVis 2013)
2 Carmen Sigovan, Chris Muelder, Kwan-Liu Ma, Jason Cope, Kamil Iskra, and Robert Ross. *A Visual Network Analysis Method for Large Scale Parallel I/O Systems.* In proceedings of 27th IEEE International Parallel and Distributed Processing Symposium (IPDPS), 2013, pages 308–319
3 Chris Muelder, Carmen Sigovan, Kwan-Liu Ma, Jason Cope, Sam Lang, Kamil Iskra, Pete Beckman, and Robert Ross *Visual Analysis of I/O System Behavior for High-End Computing.* In proceedings of 3rd Workshop on Large-Scale System and Application Performance (LSAP), 2011, pages 19–26 (Best Paper Award)
4 Chris Muelder, Francois Gygi, and Kwan-Liu Ma *Visual Analysis of Inter-Process Communication for Large-Scale Parallel Computing.* In IEEE Transactions on Visualization and Computer Graphics 15(6), 2009, pages 1129–1136 (InfoVis 2009)

4.7 Debugging and Hacking the User in Visual Analytics

Remco Chang (Tufts University, US)

The design of visualization systems always involves two primary considerations: the visualization system itself, and the user who uses the visualization. In this talk, I emphasize the consideration of the user, in particular in regards to how we could learn and leverage from the user's interactions so that the visualization system can "better understand" the user. The three examples I present are: (a) crowd sourcing past user histories using Scented Widgets [1]; (b) machine-learn the parameters of a distance metric from a user's interactions [2]; and (c) predict the user's analysis profile and behavior analyzing their interactions in real-time [3]. The three examples together demonstrate that there exist a tremendous amount of information in the user's interactions that reveal their domain knowledge, past experiences, and individual differences. For the purpose of evaluating and creating better more adaptive visualization systems, incorporating these techniques would result in more precise evaluations that could in turn lead to the design of more personalized and intuitive interfaces and systems.

References

1 W. Willett, J. Heer, and M. Agrawala. *Scented Widgets: Improving Navigation Cues with Embedded Visualizations*. EEE Transactions on Visualization and Computer Graphics, 13(6), pp. 1129–1136, 2007.
2 E. T. Brown, J. Liu, C. Brodley, R. Chang. *Dis-Function: Learning Distance Functions Interactively*. IEEE Visual Analytics Science and Technology (VAST), 83-92, 2012.
3 J. H. Zhao, Q. Lin, A. Ottley, and R. Chang. *Modeling User Interactions For Complex Visual Search Tasks*. Poster, IEEE Conference on Visual Analytics (VAST), 2013.

4.8 Re-constructing Events from Scalable and Streaming Unstructured Data

Derek Xiaoyu Wang (University of North Carolina at Charlotte, US)

URL http://cs.uncc.edu/~xwang25

Events are key elements for decision makers to form, discuss and react to social phenomena (e.g., movements, protests or campaigns) and natural phenomena (e.g. hurricane or coastal infrastructure changes). Yet, the increasing scale of digital data related to these phenomena presents challenges to sift through noisy information in order to discover meaningful events, which are characterized by changes in topics/features, geolocation, people and time. In addition, fully automated event analysis without human investigation could only provide limited evidence to decision makers. Therefore, identifying meaningful events is a fundamental challenge to visual analytics, machine learning and natural language processing research: it requires scalable architecture that balances the off-line data analytics and the interactive visual analysis with human assessment of the significance of the extracted event patterns. In this talk, I will discuss our research regarding this challenge through two novel visual analytics architectures for analyzing textual data and spatial data.First, I will introduce our scalable architecture that seamlessly integrates "big- data" text analytics (e.g. latent topic extraction and named entity recognition) with interactive visual exploration and pattern discovery. Specifically, the analytical environment presents a narrative visual interface regarding the investigative four W's: who, what, when and where for each event. It further allows users to interactively examine meaningful events using the four W's to develop an understanding of how and why. Designed to utilize Parallel Computing Clusters and other data- intensive platforms, our architecture enables stakeholders to make timely and more informed decisions with respect to on- going events, through knowing what the event is about, who is involved, where and when the event first occurred.Second, we have also applied our event-structuring concept to terrain analysis, where emergency responders must analyze temporal patterns and track changes in terrain features in response to possible natural disasters. With our narrative feature-extraction visualization platform, end-users can examine terrain features and ingest the related event structures through direct manipulations. Our platform uses search-by-example methods to automatically suggest similar terrain features and construct animations to preview the changes. The end result is not simply looking at the terrain features disjointly, but a scripted-feature analysis that can be applied to LiDAR and SONAR data at scale.Collectively, my research focuses on advancing visual analytics research to reconstruct events from unstructured data, and identify a new visual analytics design framework for improving the scale and efficiency of the problem-solving process for analyzing textual and spatial data.

4.9 Some Inspiration From Scientific Visualization . . .

Hank Childs (University of Oregon, US)

URL http://ix.cs.uoregon.edu/~hank/childs_dagstuhl.pdf

Although performance visualization will draw most heavily on techniques from information visualization, lessons learned from scientific visualization are likely useful. Specifically, the scientific visualization community has placed significant effort into flexible infrastructure that allows for dynamic composition of large numbers of algorithms. In this talk, I will discussing some of my own performance visualization endeavors to motivate why performance visualization could benefit from such infrastructure, and then present a high-level overview of the systems scientific visualization infrastructures use to achieve this user environment.

4.10 Towards an Integrated Performance Oriented Co-Design Methodology

Ulrich Rüde (Universität Erlangen-Nürnberg, DE)

Scalable algorithms are not necessarily fast, and highly optimized parallel software is not necessarily efficient. From the algorithmic perspective alone, the problems of an efficient implementation may be ignored, but just parallelizing and optimizing a given algorithm often leads also to poor results. Thus, in current practice, expensive supercomputer resources are often used inefficiently. A holistic approach will be necessary to better exploit present and future extreme scale parallel systems for computational science applications. A true co-design should incorporate the full computational modeling pipeline: It must begin where the physical model is developed, it must include the construction of the mathematical model, the discretization, the solver and of course the parallelization on all levels. Truly efficient software must equally exploit instruction level, node level, and system level parallelism. But the world of computational science has become complex and interdependent: An unfavorable choice of the discretization may inhibit vectorization, a shortsighted choice in developing the physical model may restrict parallel scalability, and an unwisely chosen solver library may lead to poor node level parallelism.

The presentation will illustrate our work towards developing a performance oriented co-design methodology for computational science and engineering together with a wish list what tool support will be necessary. The goal must be that the performance impact of any design decision along the whole pipeline should be critically quantified. This is primarily a question of a novel design methodology, but analysis tools and visualization methods are essential to make it feasible.

4.11 Scalable Visualization of Highly Distributed Computing Resources!?

Wolfgang E. Nagel (TU Dresden, DE)

ExaScale systems are expected to arrive between 2018 and 2020. However, the efficient usage of such huge and powerful computing systems is by no means trivial. The hardware gets increasingly complex, involving memory and cache hierarchies, accelerators, and hierarchical interconnects, leading to heterogeneous computing architectures. Complex parallel software will run on such hardware, composed of many complex components, including hybrid parallel paradigms, facing dynamic work loads, difficult load balancing, and more. The talk presents actual developments on the scalability of performance tools. Especially difficult gets the visualization of highest-scale performance data. The key problem is the graphical representation of a rapidly increasing number of concurrent processing elements on a display of limited size. This applies to both profile and trace data if the data has been recorded for individual processing elements. Recent research addresses the issue by means of data clustering, data hierarchies, and interactive 3D visualization.

5 Tool Demonstrations

5.1 Boxfish: a Tool for Projecting Performance Data

Todd Gamblin (Lawrence Livermore National Laboratory, US)

Joint work of Isaacs, Katherine; Landge, Aaditya; Gamblin, Todd; Bremer, Peer-Timo; Pascucci, Valerio; Hamann, Bernd
Main reference K. E. Isaacs, A. G. Landge, T. Gamblin, P.-T. Bremer, V. Pascucci, B. Hamann, "Exploring performance data with boxfish," in Companion of High Performance Computing, Networking, Storage and Analysis (SCC'12), pp. 1380–1381, IEEE, 2012.
URL http://dx.doi.org/10.1109/SC.Companion.2012.202

Understanding parallel performance data is difficult because the data is large, high-dimensional, and the relationships among measurements are not always clear. Measured data is associated with a wide variety of domains, such as network topology, multi-core node topology, logical processes and tasks, communication patterns, and the simulated physical domain. A performance problem measured in one domain may have a root cause in another domain. Boxfish is a performance data visualization tool that can project data from one domain to another according to mapping functions defined on relations of values in data domains. We demonstrate a prototype implementation of Boxfish and the power of its projection capabilities for understanding performance data.

5.2 Projections: Scalable Performance Analysis and Visualization

Abhinav Bhatele (Lawrence Livermore National Laboratory, US)

Projections is a performance visualization tool co-developed with the Charm++ parallel programming system. Performance of a message-driven, load-balanced system such as Charm++ becomes complicated to analyze. However, the same message-driven runtime can be leveraged to automatically instrument important events at very low cost. Based on such instrumentation, Projections provides tools for extensive post-mortem analysis. A distinguishing feature of Projections is the comprehensive set of views for analyzing performance that leads to insights about factors impacting performance. These include a rich timeline view and an outlier analysis tool. It also supports a novel, highly-scalable, live visualization tool for analyzing performance characteristics of running parallel programs.

5.3 Advanced Cube Features

Markus Geimer (Jülich Supercomputing Centre, DE)

Joint work of Lorenz, Daniel; Böhme, David; Mohr, Bernd; Strube, Alexandre; Szebenyi, Zoltán
Main reference D. Lorenz, D. Böhme, B. Mohr, A. Strube, Z. Szebenyi, "Extending Scalasca's Analysis Features," in A. Cheptsov, S. Brinkmann, J. Gracia, M. Resch, W. E. Nagel, (eds.), Tools for High-Performance Computing 2012, pp. 115–126, Springer, 2013.
URL http://dx.doi.org/10.1007/978-3-642-37349-7_8

In this presentation, we will demonstrate some of the more advanced features of the Cube profile browser. The main focus will be on the different views of the system dimension of profile experiments, such as the box plot and topology views showing the statistical or topological distribution of metric values across the system. In particular, the approaches implemented to display Cartesian topologies with more than three dimensions will be shown.

6 Working Groups Results

6.1 A Unified Data Model for Parallel Performance Data

Peer-Timo Bremer (Lawrence Livermore National Laboratory, US)
Bernd Mohr (Jülich Supercomputing Centre, DE)
Valerio Pascucci (University of Utah – Salt Lake City, US)
Martin Schulz (Lawrence Livermore National Laboratory, US)

Joint work of All participants of the Dagstuhl Perspectives Workshop 14022

One of the key insights for both areas – Performance Analysis and Visualization – has been that to precisely specify the problems, categorize potential methodologies, and to communicate across the area boundaries, the communities needed to establish a common unambiguous language and develop unifying models for data representation and processing. As will be discussed in more detail in the corresponding Dagstuhl manifesto, the workshop participants developed a simple yet complete framework covering the entire pipeline from data collection, aggregation, and processing to the analysis and visualization. Furthermore, for each stage of the process the new methodology allows a comprehensive classification of existing techniques and well-posed descriptions of open challenges.

The developed data model consists of a set of **spaces** with unique indices, which describe the location of a sample, a set of **metrics**, representing the values or attributes of a sample, and a number of **maps** between spaces, which induce maps between values. A space describes where in the hardware (e.g., node or core id), software (e.g., function name or call path), parallel runtime (e.g., MPI rank or thread id), application space (e.g., on which mesh elements or simulated particles), or time (e.g., wallclock timestamp or phase id) a sample data point is taken. By introducing an experiment id as another space dimension, the data model can be used to describe the data of single performance measurement experiments as well as whole experiment series. Typical metrics for HPC performance analysis are inclusive and exclusive execution time, operations counts (e.g., loads, stores, integer or floating-point arithmetic) or network packet counts, but can also be more complex and higher dimensional data points like particle coordinates, physical machine locations, or references into complex data structures.

A **measurement** is a function that for a concrete set of space indices stores a set of metric values. Given different measurement, maps between their respective spaces allow one to correlate the measurements. A typical example for a map is the mapping between MPI ranks and the node or core id on which the MPI rank executed. The map between these space – typically called the *node mapping* – enables us to understand per-core measurements, e.g., FLOP counts, with per-rank information, e.g., number of bytes send or received. Another use case for map is to model the very common technique of profile measurements where instead of keeping every single data sample, the metric values get aggregated for the whole execution of an application. This can be modeled as a mapping from the original experiment space that includes the wallclock time to a reduced space without the time dimension, where the corresponding metric values get aggregated into the sum of the values. Finally, the model incorporates semantic **contexts** to certain space, e.g., the physical layout of the hardware interconnect, which enables corresponding visual encodings, e.g., a graph layout of the interconnect. Using these basic building blocks provides a comprehensive model in which to express existing measurement approaches, processing, and post-process analysis and visualization.

Using this model not only allowed the participants to more precisely and more easily explain issues and problems to each other, but also enabled them to very rapidly identify a number of open challenges as well as cross-cutting concerns all of which will require long term in-depth collaborations between both fields. Some of the missing components are new maps between certain spaces, which are crucial to allow correlating different measurements something widely considered indispensable for root-cause analysis. Other open problems are the need for scalable data representations and processing, and new attribution techniques. On the analysis and visualization side, intuitive visual metaphors for many semantic contexts are still missing, the handling of different notions of time, e.g., wallclock vs. logical time, remains rudimentary as does the treatment of ensembles of data. Furthermore, there exist a number of general concerns such as scalability, the need for multi-resolution and adaptive techniques, and a better understanding of the concept of time in the analysis and visualization.

Participants

- Abhinav Bhatele
LLNL – Livermore, US
- Peer-Timo Bremer
LLNL – Livermore, US
- Holger Brunst
TU Dresden, DE
- Hans-Joachim Bungartz
TU München, DE
- Remco Chang
Tufts University, US
- Hank Childs
University of Oregon, US
- Todd Gamblin
LLNL – Livermore, US
- Markus Geimer
Jülich Supercomputing Centre, DE
- Judit Gimenez
Barcelona Supercomputing Center, ES
- Hans Hagen
TU Kaiserslautern, DE
- Daniel A. Keim
Universität Konstanz, DE
- Joshua A. Levine
Clemson University, US
- Naoya Maruyama
RIKEN – Kobe, JP
- Bernd Mohr
Jülich Supercomputing Centre, DE
- Christopher Muelder
Univ. of California – Davis, US
- Klaus Mueller
Stony Brook University, US
- Matthias S. Müller
RWTH Aachen, DE
- Wolfgang E. Nagel
TU Dresden, DE
- Valerio Pascucci
University of Utah, US
- Ulrich Rüde
Univ. Erlangen-Nürnberg, DE
- Carlos E. Scheidegger
AT&T Labs Research – New York, US
- Tobias Schreck
Universität Konstanz, DE
- Martin Schulz
LLNL – Livermore, US
- Derek Xiaoyu Wang
University of North Carolina at Charlotte, US
- Felix Wolf
German Resarch School for Simulation Sciences, DE