Visualizing Systems and Software Performance

Report on the GI-Dagstuhl seminar for young researchers, July 9 - 13, 2018

Organizers: Fabian Beck
Alexandre Bergel
Cor-Paul Bezemer
Katherine E. Isaacs

Abstract: This GI-Dagstuhl seminar addressed the problem of visualizing performance-related data of systems and the software that they run. Due to the scale of performance-related data and the open-ended nature of analyzing it, visualization is often the only feasible way to comprehend, improve, and debug the performance behaviour of systems. The rise of cloud and big data systems, and the rapidly growing scale of the performance-related data that they generate, have led to an increased need for visualization of such data. However, the research communities behind data visualization, performance engineering, and high-performance computing are largely disjunct. The goal of this seminar was to bring together young researchers from these research areas to identify cross-community collaboration and to set the path for long-lasting collaborations towards rich and effective visualizations of performance-related data.

Introduction

For modern systems and software, it is of utmost importance to ensure a high level of performance. Performance issues can have catastrophic consequences. For example, a performance issue led to the outage of healthcare.gov[1], denying hundreds of millions of American citizens access to the health insurance system. Performance issues can also have large monetary consequences. For example, it is estimated that a 1-second slowdown of Amazon costs $1.6 billion per year[2].

To prevent performance problems and ensure good performance of a system or software, it is necessary to continually monitor and analyze performance-related data. However, as the scale at which systems and software operate increases, the scale of performance-related data increases as well, making the analysis of this data challenging. One solution for analyzing large-scale performance data is to make use of visualization.

Visualization of large-scale performance data is a topic that is studied by the performance engineering, software visualization, and high-performance computing communities. However, these research communities are mostly disjoint and mutually beneficial knowledge is rarely shared between them. In addition, the development of a new visualization technique is an iterative process, which requires continual back-and-forth feedback between visualization and performance researchers (or practitioners) to evaluate the applicability of the technique to a real-world problem. The gap between research communities hinders the evaluation of new techniques.

Performance Engineering (PE) is the field that integrates theory and practice around the topic of benchmarking, performance evaluation, and experimental system and software analysis in general. It considers both classical performance metrics such as response time, throughput, scalability and efficiency, as well as other non-functional system properties such as availability, reliability, and security. The research effort provided by the PE community spans the design of metrics for the evaluation as well as the development of methodologies, techniques and tools for measurement, load testing, profiling, workload characterization, dependability and efficiency evaluation of large-scale computing systems and software.

Software Visualization (SV) is a broad research area encompassing concepts, methods, tools, and techniques that assist in a range of software engineering and software development activities. Of particular interest in performance is the development and evaluation of approaches for visually analyzing software and software systems, including their structure, execution behavior, and evolution. The research effort provided by the SV community comprises effective visual metaphors and interaction techniques to cope with analyzing and understanding multiple aspects of complex systems.

High-Performance Computing (HPC) is a research area encompassing the technology and knowledge necessary to compute much larger problems than can be run on a personal ma-

1 Amber May. What went wrong with HealthCare.gov – and what now. http://www.benefitspro.com/2013/11/20/what-went-wrong-with-healthcaregov-and-what-now, 2013. (last visited: June 15, 2017).

2 Kit Eaton. How one second could cost Amazon $1.6 billion in sales. http://www.fastcompany.com/1825005/how-one-second-could-cost-amazon-16-billion-sales. (last visited: June 15, 2017).

chine. As such, HPC spans a wide range of topics including computer architecture and networking, GPUs and accelerators, programming languages and compilers, cloud and distributed technologies, data storage and I/O, parallel algorithms, and energy efficiency. Performance is of great interest in HPC as even a small increase in efficiency can save significant amounts of supercomputing time than can then be allocated to another application.

The main goal of the seminar was to bring together young researchers (PhD students, postdocs or assistant professors) in the areas of (i) performance engineering, (ii) software visualization, and (iii) high-performance computing. Each field presented their current research efforts, and exchanged experience and expertise, discussed research challenges, and together developed ideas for future collaborations. In particular:

- The seminar provided a joint forum for participants coming from different, mostly disconnected research communities.
- The participants learned about the latest developments in their own as well as adjacent research areas.
- The seminar fostered interaction among the participants and will likely establish collaborations between the researchers towards joint research projects.

Seminar Schedule and Results

The schedule of the seminar spanned a complete week, from July 9 to July 13, 2018. The seminar comprised joint sessions, group workshops, as well as opportunities for interactions on individual level. The schedule included tutorial and demonstrations, breakout groups, and individual and grouped collaborations. This section summarizes the outcomes of the different parts and sessions, while a detailed schedule is part of the appendix.

Tutorials

Three 60-90 minute volunteered tutorials in plenary sessions provided insights into the different communities as well as showed their practices for data recording and visualization. They covered performance engineering, high performance computing, as well as scientific visualization. An important role of the tutorials was that they laid a foundation for a joint language and understanding of each others fields. They also provided entry points for starting to work with data or visualizations tools used in the different communities.

- *State of the Art of Visualization in APM Tools* (André van Hoorn & Dušan Okanović)
- *Tracing and Profiling in HPC* (Marc-André Hermanns & David Böhme)
- *Scalable HPC Visualization and Data Analysis Using VisIt* (Kevin Griffin)

Tool and Data Demonstrations

During the seminar, we had six informal tool demonstrations. During these tool demonstrations, participants gave a brief overview of their tools, often with a short demonstration. The goal of these ad-hoc tool demonstrations was to make other participants aware of the state-of-the-art in performance visualization. The demonstrations were filled in when there was a gap in the schedule.

- *Java Profiling in the Code Editor* (Oliver Moseler)
- *Using Spectrum Analysis to Identify Meaningful Views in Call-Path Performance Profiles* (Tom Vierjahn)
- *Prototype that was used for: What do Constraint Programming Users Want to See? Exploring the Role of Visualisation in Profiling of Models and Search*[3] (Sarah Goodwin)
- *Visualizing Object Factories* (Juan Pablo Sandoval)
- *Various Kinds of HPC Data* (Abhinav Bhatele)
- *Pervasive Visualization in Augmented Reality for Software Monitoring*[4] (Leonel Merino)

Breakout Groups

The breakout groups focused on specific topics and problems for visualizing system and software performance. In a first phase of breakout groups in the first half of the week, we selected topics that arose during the initial brainstorming and group discussions on Monday. A second phase followed where the topics of the first phase were refined or new groups were formed around new questions that came up during the seminar. Each group decided at the beginning what outcome they target and assigned roles for a moderator and a scribe. The outcomes of the groups varied from identifying research challenges and designing visualization solutions to a hackathon with an interactive visualization prototype.

The goal was to keep these breakout groups small (4-5 participants), but in the end several groups merged leading to larger groups. We did not feel the size of these larger groups was an issue. At several points during the seminar, we asked the breakout groups to give an informal update, so that (i) other groups were aware of the progress of the group and (ii) participants had the opportunity to switch between groups based on their expertise and interest. As a result, we noticed that several participants who have broad research interests or expertise participated in several of the breakout groups. Short presentations at the end of the first and second phase document the outcomes of the groups, together with the written abstracts presented in the following.

Code Mining & Refactoring

Moderators: Philipp Leitner, Juan Pablo Sandoval, Santiago Vidal (role switched)
Scribes: Jinfu Chen, Diego Costa, Juan Pablo Sandoval (role switched)
Members: Davide Arcelli, Cor-Paul Bezemer, Katherine Isaacs, André van Hoorn

Application source code is a living entity, changing due to various factors such as refactoring, bug fixes, and the addition of features and documentation. The effect of these changes on

3 Goodwin, S., Mears, C., Dwyer, T., de la Banda, M. G., Tack, G., & Wallace, M. (2017). What do constraint programming users want to see? Exploring the role of visualisation in profiling of models and search. IEEE Transactions on Visualization and Computer Graphics, 23(1), 281-290. https://doi.org/10.1109/TVCG.2016.2598545

4 http://scg.unibe.ch/wiki/projects/mastersbachelorsprojects/pervasive-visualization-in-ar-for-software-monitoring

performance, and on other factors such as code maintainability, is of interest to both practitioners and researchers. The breakout group first discussed possible designs of visualizations that might help practitioners relate performance changes to types of code changes, coming to an agreement that basic statistical charts found in dashboards are enough at this high level of abstraction. However, close inspection to discover why the code changes caused the performance changes is difficult and visualization may help. The close inspection moves from the performance-centric line charts to code-centric graphs and code views. State-of-the-art visual support usually involves comparing the calling context trees or call graphs of two commits. The code changes can results in significant structural differences in these graphs and graph comparison in general is an open visualization problem, but using extra semantics from software may help. Questions about which versions to analyze remain.

The breakout group also proposed combining performance metrics with repository views, such as the Git branching graph. Beyond visualization, several members of the breakout session had experience mining various anti-patterns such as performance anti-patterns and modifiability anti-patterns. Mining for multiple categories of anti-patterns and ranking them by severity was suggested as a future topic of research, aimed at studying trade-offs between such categories.

One challenge of research in this area is the lack of good data sets to work with. Few projects do performance regression testing, limiting the applicability of possible research at this time. The breakout group considered developing a benchmark or data set to encourage research in this area. Benchmarks could be run post-mortem along a commit history as synthetic data.

Comparative Visual Analytics of Performance Data

Moderator: Marc-André Hermanns
Scribes: Abhinav Bhatele, Patrick Gralka, Kevin Griffin (role switched)
Members: Fabian Beck, David Boehme, Angelina Lee, Olga Pearce, Dušan Okanović, Paul Rosen, Luka Stanisic, Tom Vierjahn

Visual analytics combines mining techniques and algorithmic analysis with visualization into an iterative process. In the first phase of the breakout group, the group discussed how such combinations of algorithms and visualization can be leveraged in performance engineering (for HPC and other applications), with a specific focus on comparative analysis. Such comparison is important to analyze historical execution data (compare points in time) of an application or a system, to contrast different executions, or to find similarities within a single execution. The main challenge is the comparison of call tree and graph structures. Work on visual hierarchy and network comparison provides a basis for such approaches, but the comparison of execution data is particularly difficult because of the scale and variance of the data. Whereas some performance visualization approaches are already working towards this direction, an open question is how to further scale these approaches, integrate further data sources, and apply appropriate analysis techniques to filter and abstract the data. As a roadmap for future work, the group suggests to survey in more detail existing visualization approaches, and finally developing a tailored visual analytics solution for a relevant performance engineering use case.

In the second phase, the group focused on designing a specific visualization for the comparative performance analysis of call context trees and graphs. The group discussed metrics and dimensions that are relevant for such visualization. Important tasks for developers that should be supported are to find outliers, to compare trees or subtrees as well as to find bottlenecks and relate these to the source code. Challenges that need to be solved to this end include handling the scale of the data, providing good interactions for data exploration, and representing structural differences in call trees. Also, the process of performance analysis is tool and problem specific, mostly not documented but only an implicit process known by the domain or performance expert. The group designed a call graph representation (Figure 1) that might form a basis for further exploring and solving these challenges. As future work, the group considers to sketch different visual solutions, work together with domain experts to evaluate these, and finally develop an integrated visual analytics systems that links the performance data and source code.

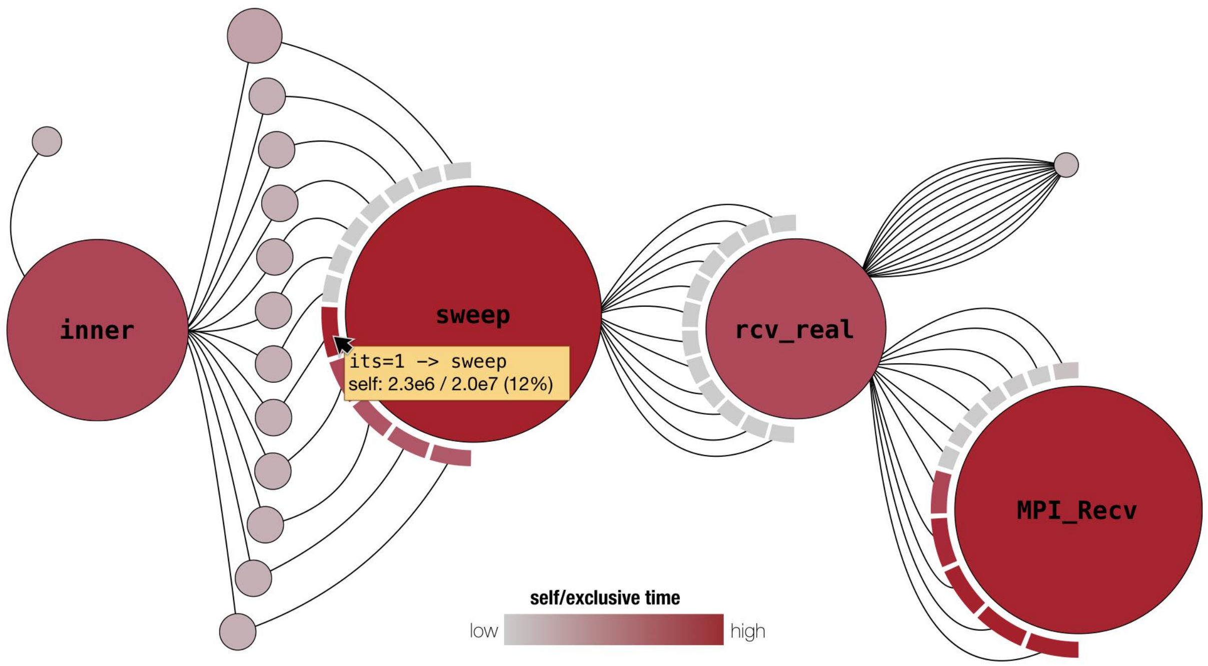

Figure 1: Call graph representation visualizing several metrics in nodes and edges.

Conducting User Studies for Evaluating Performance Visualizations

Moderator: Oliver Moseler
Scribe: Sarah Goodwin
Members: Fabian Beck, Alexandre Bergel, Cor-Paul Bezemer, Diego Costa, Philipp Leitner, Leonel Merino, André van Hoorn

Evaluating performance visualization tools is recognized to be a daunting task as many researchers experience. Once a visualization is designed, understanding the real implication for a user is complex.

This breakout group focused on evaluating visual tools on performance visualization. The group focused on three different tasks: (i) understanding what *evaluation* means in different communities; (ii) producing a survey to gather the most relevant challenges experienced by the participants of the seminar; (iii) defining the foundation of a larger effort to be carried out after the seminar about understanding the requirements when visualizing performance-related data.

The survey among the seminar participants with 20 responses showed that 75% of the researchers had already evaluated a performance visualization, but only 25% in form of a user study (all of the latter from the software engineering or visualization community). Regarding their confidence to perform user studies, only 20% expressed high confidence. The main challenges for evaluating performance visualizations (identified in a group coding session of free-text answers), the participants considered the following (frequency in brackets, only listing challenges mentioned more than two times):

- Defining appropriate evaluation tasks (8)
- Finding appropriate participants from the target group (6)
- Integration with real-world situations (tools, workflow, scalability) (5)
- Defining a meaningful and/or fair baseline to compare to (5)

The group suggests for future work to re-conduct the survey study with more participants and additional questions. A goal of this effort will be to derive guidelines for pitfalls and best practices on evaluating software performance visualization. Also, empirical data on what performance analysis tools are used in which context by researchers and practitioners would be helpful for identifying appropriate baselines. Structuring performance engineering tasks and contextualizing them with respect to software development stages will reveal scenarios that can potentially be supported by visualization.

An HPC Visualization Hackathon

Members: Alexandre Bergel, Marc-André Hermanns, Santiago Vidal, Tom Vierjahn

During the seminar a “hackathon” was conducted. The initial goal was to quickly build a visualization from the ZEUS benchmark result[5] provided by Marc-André Hermanns. The programming effort was balanced among the participants of the hackathon. Such effort was carried out to extract the dataset and to provide them in a parseable format, then to import them, and to visualize them. The Roassal visualization engine was used for that purpose. The produced visualizations (Figure 2) can form a base for a joint collaboration.

5 David Böhme, Markus Geimer, Felix Wolf, & Lukas Arnold. (2018). Scalasca analysis report for SPEC MPI.2007 benchmark 132.zeump2 on 512 processes in virtual-node mode on Blue Gene/P [Data set]. Zenodo. http://doi.org/10.5281/zenodo.1211448

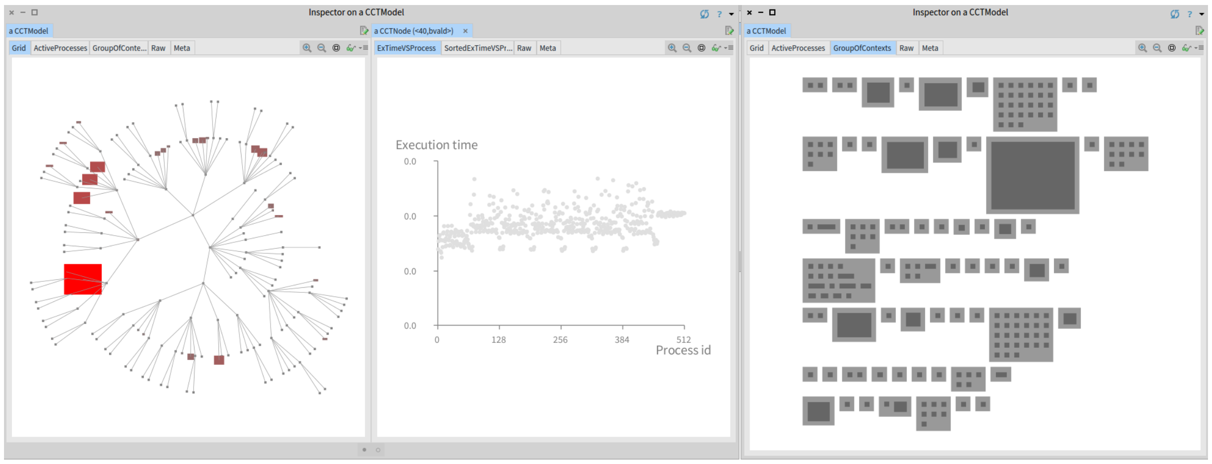

Figure 2: Visualizations of data from the ZEUS benchmark produced during the hackathon.

Collaboration Rodeo

Moderator: Cor-Paul Bezemer
Scribe: Kate Isaacs
Members: Abhinav Bhatele, Jinfu Chen, Kevin Griffin, Paul Rosen, Juan-Pablo Sandoval, Luka Stanisic

The purpose of this breakout group was to sit together with performance regression detection experts and high performance computing experts and have a Q&A session about (i) the type of problems that HPC experts have, (ii) how existing performance regression detection approaches can help with these problems, and (iii) the identification of what related performance data is available in HPC systems. The group discussed and laid groundwork in four areas for collaboration.

First, members of the HPC community are starting to collect regular performance regression data. As this area is largely new to the HPC community, they may leverage existing work from the Software Engineering and Performance Engineering communities. The scale, concerns, and workflow of this project make it a prime opportunity for an informative case study on performance regression.

Second, the HPC community is also starting to collect machine-wide data from temperature sensors within the machine room, to the performance of particular applications, to the mix of applications running on a shared cluster, to network traffic data, to performance counters on the individual nodes, to local power and temperature readings. Such a wide variety of data has not been available before and mining for correlations related to performance behavior is an open problem.

Third, it came to light during the tutorials that VisIt[6], a scientific visualization platform, can collect detailed usage logs. Users of VisIt can turn these logs into scripts for later re-use while developers of VisIt can use these logs for debugging. Mining this available log data may yield an understanding of the visualization workflow in HPC and yield improvements in usability of VisIt as a tool.

6 https://wci.llnl.gov/simulation/computer-codes/visit

Fourth, performance debugging often relies on the intuition of performance experts. Developers without such expertise may not know where to begin the performance debugging process. The group discussed possibilities in making performance debugging more widely accessible. Logging and mining the procedures of experts could possibly result in knowledge and recommendations to aid those without experience. Unlike the previous three topics, such data is not presently available.

Conclusion

The seminar was a highly productive and fruitful meeting as important research challenges have been identified, participants learned about related fields, and a number of research efforts have been initiated. Academic venues considered for the disseminating the efforts are ICPE (ACM/SPEC International Conference on Performance Engineering), VPA (International Workshop on Visual Performance Analysis), and VISSOFT (IEEE Working Conference on Software Visualization). To enable a well-defined process of sharing data between the involved communities, the participants created virtual community VSSP on Zenodo[7]. The participants will further explore options to continue VSSP as a workshop series.

7 https://zenodo.org/communities/vssp/

Appendix

Schedule

Monday, July 9

9:00	Intro & overview
9:30	Icebreaker
10:30	Break
11:00	Research lightning intros
12:00	Lunch
14:00	Breakout groups topic selection
14:30	Informal topic selection presentation by the moderators of each group
15:00	Define groups and group deliverables for Tuesday
16:30	Share breakout discussion (and choose group for Tuesday)
17:00	Free time
18:00	Dinner

Tuesday, July 10

9:00	Tutorial: State of the Art of Visualization in APM Tools (André van Hoorn & Dušan Okanović)
10:00	Break
10:30	Breakout groups
12:00	Lunch
14:00	Breakout groups
17:00	Free time
18:00	Dinner

Wednesday, July 11

9:00	Tutorial: Tracing and Profiling in HPC (Marc-André Hermanns & David Böhme)
10:30	Break
11:00	Breakout reports
12:00	Lunch
14:00	Excursion to Trier
18:00	Dinner

Thursday, July 12

9:00	Tutorial: Scalable HPC Visualization and Data Analysis Using VisIt (Kevin Griffin)
10:00	Break
10:30	Second round breakout groups
12:00	Lunch
14:00	Second round breakout groups
16:30	Breakout writing
17:00	Free time
18:00	Dinner

Friday, July 13

9:00	Breakout reports
10:00	Break
10:30	Future planning, feedback & wrap up
12:00	Lunch

List of Participants

- Davide Arcelli (University of L'Aquila, Italy)
- Fabian Beck (University of Duisburg-Essen, Germany)
- Alexandre Bergel (University of Chile, Chile)
- Cor-Paul Bezemer (University of Alberta, Canada)
- Abhinav Bhatele (Lawrence Livermore National Laboratory, US)
- David Boehme (Lawrence Livermore National Laboratory, US)
- Jinfu Chen (Concordia University, Canada)
- Diego Costa (Heidelberg University, Germany)
- Sarah Goodwin (Monash University, Australia)
- Patrick Gralka (University of Stuttgart, Germany)
- Kevin Griffin (LLNL/University of California, US)
- Marc-André Hermanns (Jülich Supercomputing Centre, Germany)
- Katherine Isaacs (University of Arizona, US)
- Angelina Lee (Washington University, US)
- Philipp Leitner (Chalmers, Sweden)
- Leonel Merino (University of Bern, Switzerland)
- Oliver Moseler (University of Trier, Germany)
- Dušan Okanović (University of Stuttgart, Germany)
- Olga Pearce (Lawrence Livermore National Laboratory, US)
- Paul Rosen (University of South Florida, US)
- Juan Pablo Sandoval (Universidad Catolica Boliviana, Bolivia)
- Luka Stanisic (Max Planck Computing & Data Facility, Germany)
- André van Hoorn (University of Stuttgart, Germany)

- Santiago Vidal (UNICEN University, Argentina)
- Tom Vierjahn (RWTH Aachen, Germany)