Graphs Are Not Enough: Using Interactive Visual Analytics in Storage Research

Zhen Cao,[1] Geoff Kuenning,[2] Klaus Mueller,[1] Anjul Tyagi,[1] and Erez Zadok[1]
[1]*Stony Brook University* and [2]*Harvey Mudd College*

Abstract

Storage researchers have always been interested in understanding the complex behavior of storage systems with the help of statistics, machine learning, and simple visualization techniques. However, when a system's behavior is affected by hundreds or even thousands of factors, existing approaches break down. Results are often difficult to interpret, and it can be challenging for humans to apply domain knowledge to a complex system. We propose to enhance storage system analysis by applying *interactive visual analytics*, which can address the aforementioned limitations. We have devised a suitable Interactive Configuration Explorer (ICE), and conducted several case studies on a typical storage system, to demonstrate its benefits for storage system researchers and designers. We found that ICE makes it easy to explore a large parameter space, identify critical parameters, and quickly zero in on optimal parameter settings.

1 Introduction

Analyzing and understanding the behavior of storage systems has always been of interest to researchers and system administrators. Previous work has presented analysis of various aspects: performance [8, 15], reliability [20], energy consumption [31], etc. In recent years, storage systems have grown more complex with the addition of new hardware, varied workloads, and increasing scale. This makes analyzing storage systems more important but also more challenging.

Both *non-visual* and *visual* techniques have been applied in analyzing storage system behavior. Non-visual methods include statistical measurements such as mean, standard deviation, and percentile(s) [12], plus machine-learning techniques including classification and clustering analysis [3,42]. Visual approaches have included 2D techniques such as histograms [19], box plots [8], etc., and 3D versions such as surface plots [10, 38].

However, existing techniques are not enough for thorough understanding of storage system behavior, for three reasons. First, *storage systems are often impacted by many factors*. Modern storage systems can easily have hundreds of tunable parameters [9]. However, most commonly applied visualization techniques (e.g., line, histogram, scatter plots) can focus on one or few factors within one plot. To analyze the impact of all parameters, multiple figures are needed. For example, during our previous study of just nine parameters in a typical storage system [9], we produced over 2,000 plots in an attempt to fully analyze the parameters' impact and dependencies. The problem is exacerbated because specific workloads and the underlying hardware can also affect system behavior [8, 9, 31]. Moreover, some storage parameters have categorical values, while many plotting approaches (line, scatter, etc.) assume numerical axes. The standard regression technique of splitting categorical parameters into dummy binary values does not scale well, because it makes the configuration space grow exponentially [41].

Second, *some traditional approaches lack interpretability*. Storage researchers often want not only to explain the numbers, but also to understand the underlying implications at the system level. Many existing approaches project high-dimensional data into low-dimensional spaces; the newly constructed dimensions are usually linear or nonlinear combinations of the originals. Examples include Principal Component Analysis (PCA) [32], Independent Component Analysis [17], and visual techniques such as Multi-Dimensional Scaling (MDS) [23]. One major drawback of these approaches is that physical meaning of each dimension may not be preserved after projection [24].

Third, *it is difficult to infuse domain knowledge*. It is important and beneficial to combine expert knowledge into analysis procedures. For example, in our previous study we used our storage expertise to pick nine representative storage parameters and four common workloads [9]. Similarly, Basak *et al*. [3] pre-selected features manually when doing workload characterization. Due to the complexity of storage systems, there is no single master solution that can satisfy all requirements; often a combination of statistics, visualization, and human reasoning must be applied. However, current storage papers mostly use static, non-interactive 2D (occasionally 3D) plots, which make it inconvenient to exploit domain knowledge while analyzing.

To address the aforementioned limitations, we propose to apply another type of analytic technique in storage research: **interactive visual analytics**. Interactive visual analytics can often present high-dimensional spaces in a single 2D space, allowing researchers to explore interactions among multiple factors of the targeted system. They let users exploit their domain knowledge and intuition via visual interaction; this empowers users to take an active role in the analysis process, better understand the target system, and make sound decisions with high confidence.

To demonstrate the benefits of applying interactive visual analytics, we took storage-system performance analysis as an example. We conducted studies on our three-year dataset

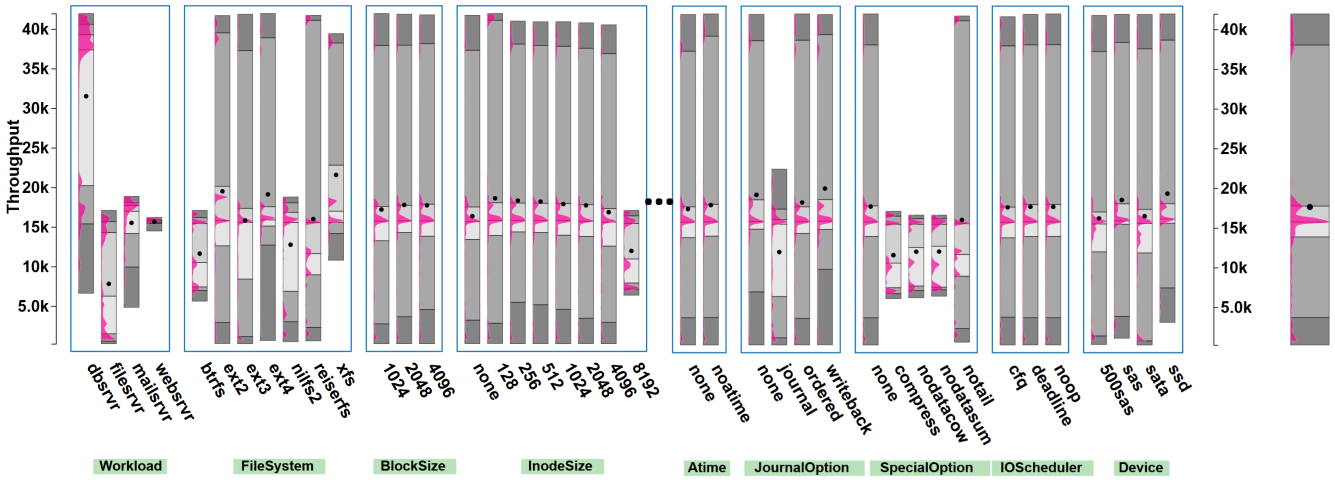

Figure 1: Screenshot of ICE. Block Group was cropped out, shown as "…" in the figure, to increase text readability.

collected on a typical storage system; the dataset has 9 dimensions and 100k configurations (about 500k data points in total); many large installations often collect numerous similar telemetrics [26, 27, 30]. We prototyped a new tool, the *Interactive Configuration Explorer* (ICE), which uses an enhanced box plot whose form is well understood by storage researchers, with an embedded density plot for throughput distribution, to present the data to the user in a compact, easily interpretable form. We have found that ICE can help researchers explore the interaction among multiple storage parameters (numeric, discrete, and categorical), and understand system performance, efficiency, stability, reliability, etc. We hope our study and this paper will lead to more use of interactive visual analytic approaches in storage research.

Related Work. Interactive visual analytics have been applied to help explore and understand many real-world datasets. Existing tools and approaches include plotly [33], Tableau [35], Parallel Coordinates [18], Parallel Sets [21], and Data Context Maps [11], etc. Rodeh *et al.* [29] visualized block I/O workloads on a two-dimensional grid, to help analyze measurements such as read/write ratios and sequentiality, and to catch hit-rate changes over time. Interactive visual analytics have been shown to be useful in other types of system analyses, including network traffic [2,13,14,28,40] and database query optimization [4, 37]. Due to the nature of storage parameters, however, many existing interactive visual analytic techniques may not be directly applicable. We detail their limitations in Section 2.

2 ICE: Interactive Configuration Explorer

A motivating example. Maria is an analyst responsible for a large storage system. She has been working on a performance problem for weeks, without success. Fortunately she has a testbed for benchmarking the system and has collected lots of data about different configurations and workloads. But she needs to make sense of all those numbers, which is what our interactive visual analytic tool, the *Interactive Configuration Explorer* (ICE), is designed to do. Launching it, Maria first sees Figure 1. The metric that matters to her is throughput (Y axis—higher is better). Her system is currently used as a file server, so she clicks on "Workload", fixing it to "fileserver", and the screen reconfigures to show just the file-server data (Figure 2, zoomed to show only the first two sections). The mean performance of each filesystem is shown by the black dots, and the range by the length of the bar. Maria sees that both *btrfs* and *xfs* have high throughput, but *xfs* has less variance. Nevertheless, she decides to look further into *btrfs* because of its snapshoting capabilities. Choosing that option produces Figure 3, where she chooses an 8KB inode size for its low variance, and sees that selecting *compress* for the "SpecialOp" will reduce variance further.

Maria knows that the system might later be used as an OLTP database server. Will *btrfs* still behave well? She backs out, selects *dbserver*, and sees Figure 4. It turns out that Btrfs is terrible for database workloads. But Ext4 seem to contain some high-throughput configurations (indicated by the peaks in the magenta regions inside the bar). She can now use ICE to select *ext4* and again explore different parameter selections for the *database* workload.

Both researchers and administrators commonly encounter this scenario: analyzing storage systems that are impacted by tunable parameters and other factors including workload, hardware, software, etc. As in the example, interactive visual analytics allow quick exploration of many configuration options. We now describe the design of ICE, and in Section 3 we will show examples of how we used it to understand storage system behavior.

Data collection. ICE evolved from an earlier project [9] where we collected a large amount of experimental data on 7 different file system types and 4 representative workloads using Filebench [1, 39]. We experimented with *block size*, *inode size*, *blocks per group*, *mount option*, *journal option*, *special option*, *I/O scheduler*, and with 4 different storage devices. There were 24,288 unique configurations, and we

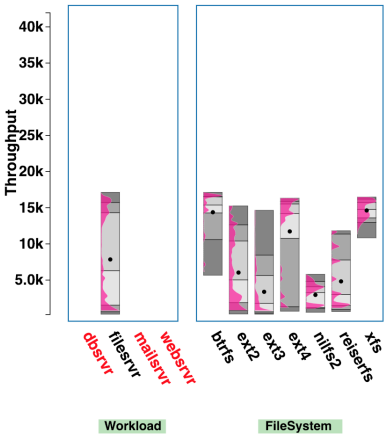

Figure 2: Partial screenshot of ICE after selecting the "fileserver" Workload.

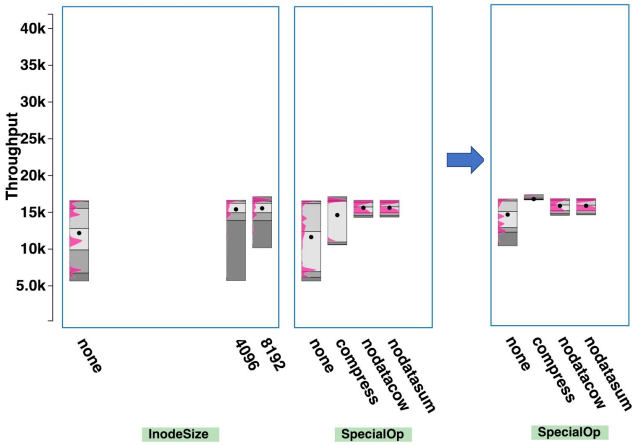

Figure 3: Using ICE to select parameter values for btrfs under the fileserver workload (partial screenshots).

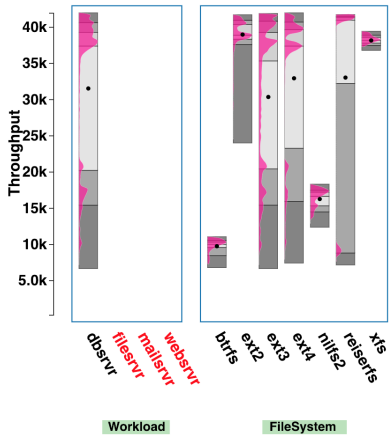

Figure 4: Partial screenshot of ICE after selecting the "dbserver" Workload.

collected more than 500,000 data points over 3 years [7]. Our datasets now consist of 10^{+} dimensions, i.e., tunable file system and kernel parameters, hardware devices, and workloads, which makes analysis challenging when using only traditional analytic approaches, due to the limitations discussed in Section 1. This motivated us to develop ICE.

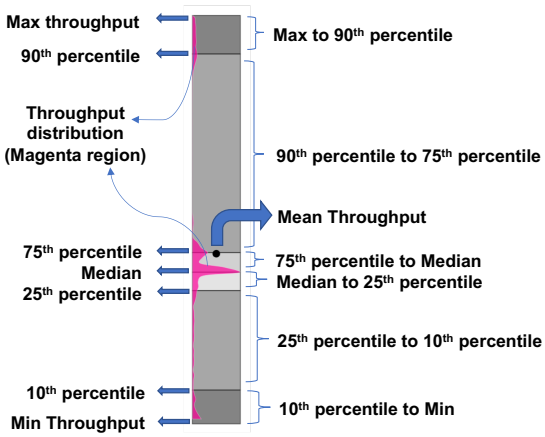

Figure 5: Annotated bar plot explaining how to read it.

Design of ICE. We started by applying existing popular interactive visual analytic approaches, such as Parallel Coordinates [18] and Parallel Sets [21]. We found that many of them were not directly applicable. For example, the Parallel Coordinates approach lists and orders the values of one parameter along one axis, and thus poorly supports categorical parameters (common in storage systems). Moreover, these tools can sometimes be complex and difficult for storage researchers to use, requiring a relatively long training period. Therefore, one design principle of ICE is usability.

ICE shows various configurations as bars, each of which is carefully designed to present rich information about the throughput distribution resulting from selected parameter values. Bars in ICE are a combination of stacked bar plots and *violin* plots [16], which are box plots superimposed with rotated kernel density plots. Figure 5 shows an annotated example of one such bar. The shading distinguishes different throughput percentiles: the darker shades on the top and bottom represent the range from the maximum to the 90th percentile and from the minimum to the 10th percentile. Medium shades mark the ranges for the 90th to 75th and the 10th to 25th percentiles; the lightest shades in the middle mark the 75th to 25th percentiles. The black horizontal lines in each bar mark major percentile boundaries (90th, 75th, 50th (median), 25th, and 10th). The mean of the data is indicated by a solid black dot. In addition, the distribution of the data is shown by the magenta-colored area(s) on each bar, giving more detail about how the configurations represented by that bar are distributed in the throughput space. We chose all colors and shades carefully by using ColorBrewer [5, 6], ensuring that they are visible on a variety of displays and to users who suffer from color-blindness.

Returning to Figure 1, ICE is inspired by *scented widgets* [43], which were originally proposed as graphical user interface controls enhanced with embedded visualizations that facilitate navigation in information spaces. We see that ICE displays multiple bars grouped by parameters, with each

bar representing the throughput distribution of a subset of configurations in which one parameter is fixed to a given value. Since ICE is interactive, all of these bars change as the user explores the data. Note that some parameter types have been omitted for space reasons; a full version of the display can be seen at *http://www.fsl.cs.stonybrook.edu/%7Ezhccao/ice*. A cumulative bar at the right-hand side of the figure, which also changes during exploration, shows the throughput distribution of the union of the chosen configurations. In this example, the initial display shows the distribution of all configurations for 7 file systems across 4 workloads.

Given an initial setting, the user can select any combination of workload, file system, and storage parameters, and the bars will be updated to show throughput distribution, as we saw in the example above. With this design, users can easily select parameters with different objectives. For example, Maria maximized throughput by selecting the bar with the highest solid black dot, but she could also reduce performance variance by focusing on shorter bars. Our case studies in Section 3 show how ICE can help users make such configuration decisions.

ICE is able to display selected datasets in real time, taking less than 2 seconds per update in our case. It was designed for generic datasets collected on different storage systems. We plan to make ICE open-source to facilitate research on understanding storage parameter spaces and optimizing large storage systems.

3 Case Studies

In Section 2 we showed one example of using ICE to analyze storage system throughput and tune parameters to achieve high performance. In this section, we describe two more case studies to show how ICE can also help analyze performance stability and reliability. These studies are based on our real experience in analyzing and tuning storage systems [9, 44].

3.1 Performance Stability

Now suppose that Maria wants to configure a system as an email server, for which she cares about performance stability. The *range* (difference between maximum and minimum) and *Inter-Quartile Range* (IQR) (difference between 75th quartile and 25th quartile) are often used to quantify stability in system performance [8]. ICE visually presents the range as the length of each bar, and IQR as the length of the lightest shade in the middle of each bar (see Figure 5). Maria starts her analysis with ICE and selects the *mailserver Workload*. The left part of Figure 6 shows a partial ICE screenshot after doing so. The bars for each parameter value present the updated throughput distribution if that value is chosen. For example, the bar above "btrfs" shows the throughput distribution of all Btrfs configurations under the mailserver workload. The updated bars guide Maria to select a value for another parameter, based on her objectives. Clearly, under *mailserver* *xfs* has by far the smallest throughput range, even though its

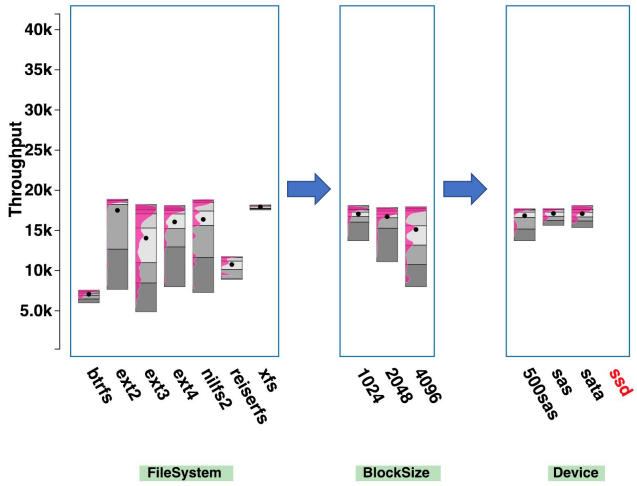

Figure 6: Using ICE to optimize a mail server (partial screenshots). We chose Workload = "mailserver"; FileSystem = "ext4"; and BlockSize = "1024".

highest throughput value is slightly lower than those of *nilfs* and *ext2*. Since stability is the primary concern, Maria thus decides to configure her server using XFS.

Unfortunately, Maria's boss informs her that upper management has established a policy that all corporate computers have to use the Ext4 file system, regardless of the application. She returns to ICE, selecting *ext4*. Since Ext4 shows a wide range of throughput, indicating unstable performance, she now continues configuring its parameters. As shown in the middle part of Figure 6, a value of *1024* for *BlockSize* gives the most stable result; Maria thus chooses this value.

The right part of Figure 6 shows the final step. Maria has three types of HDDs available: "sas" (a 146GB SAS HDD), "500sas" (a 500GB SAS HDD), and "sata" (a 250GB SATA HDD). She estimates that her email system will only need 100GB, so she can ignore the HDD capacity and focus solely on performance. The bar associated with *sas* appears the the shortest, which means the 146GB SAS HDD has the most stable performance. Therefore, Maria selects that HDD.

3.2 Constrained Tuning

When analyzing storage systems, multiple objectives sometimes need to be considered at the same time. For example, system administrators may want to configure a *stable* system (i.e., low variability) and still achieve high *performance*. In this case certain constraints may be added to the analysis process. Here we show an example of how ICE can be easily applied to reflect such constraints and help the analysis.

This time Maria wants to configure her system as an OLTP database server that uses Ext4. However, she wants to ensure reliability for the file system; therefore, she sets the Ext4 journaling mode to *data=journal*. She then uses ICE to analyze the system and help her find the configuration that leads to the highest throughput under the current constraints, as shown in Figure 7. She has already tested four device

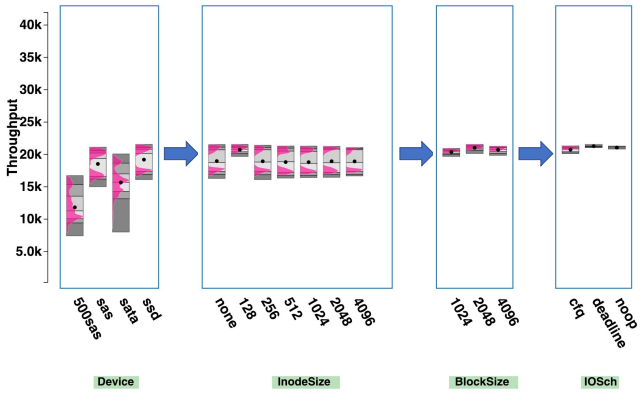

Figure 7: Using ICE to optimize multiple constraints (partial screenshots). We chose Workload="dbserver"; FileSystem = "ext4"; Device = "ssd"; InodeSize = "128"; BlockSize = "2048".

types, 3 HDDs and 1 SSD. Unsurprisingly, the SSD shows the highest maximum throughput (top of bar) so she chooses that. ICE then updates the rest of the display to reflect that choice, and Maria then focuses on the *Inode Size*. A 128-byte inode is clearly preferable in this situation, so she selects that and moves to the *Block Size* where 2KB has better stability. Finally, she chooses *deadline* for the *I/O Scheduler*, as it improves stability more without hurting performance, as shown in the rightmost part of Figure 7.

It is important to note that Maria is not restricted to following the particular path given in this example. She could have chosen an I/O scheduler first, followed by selecting the best block and inode sizes, and waited to the end to chose a disk type. She also could have selected an inode size, observed how it interacted with the other parameters, and backed out so that should could pick a value later based on her choices for something else. One of the benefits of ICE is that the user can easily and quickly try different options to see how it affects the results, exploring the parameter space along the path that best suits her needs and research style.

4 Future Work

This position paper advocates the use of interactive visual analytics for storage systems analysis and research. We plan to continue this work. In particular, the following three enhancements to ICE are promising: (1) During interactive analysis, it is useful to track the progression of the analysis and how the current state compares to previous ones. We plan add a provenance scheme [25, 34] that would show previous results along a timeline, enabling us to see the "path" by which a particular analysis was reached. (2) ICE is already scalable in the number of presented *configurations* since it displays distributions, and new datasets can be dynamically imported. We plan to further improve ICE to support even larger spaces, consisting of hundreds or thousands of parameters. Previous work has demonstrated that some parameters have greater impact than others [9, 41]. We plan to expand ICE to visualize and help analyze parameter importance based on measures of redundancy, uniqueness, coverage, mutual information [9], etc. (3) ICE was designed generically for storage-system analysis, where categorical parameters are quite common. It can already be used to explore many metrics, such as stability, I/O latency, etc. We plan to further extend ICE so that users can explore multiple metrics simultaneously. For instance, users might want to achieve high throughput while maintaining relatively low energy consumption. (4) ICE evolved from our efforts to analyze and auto-tune storage system performance. We collected large amounts of data and ICE successfully helped our analytic work. Since ICE can easily visualize the importance and correlation of parameter values, we plan to investigate how ICE, fed with a small amount of data initially, can guide a further data collection process by suggesting promising configurations to experiment with. (5) We designed ICE to analyze computer system parameter spaces, where some previous techniques have not proven as useful as one might wish. Nevertheless, we are investigating ways to incorporate approaches such as Parallel Coordinates [18], Parallel Sets [21], and Data Context Maps [11] into ICE. We also would like to integrate machine-learning techniques [22, 36, 41] to help guide the analyst in exploring large parameter spaces.

Lastly, we will open-source ICE to benefit the storage research community and hopefully lead to more work on this new but promising area of interactive visual analysis.

5 Conclusions

The Interactive Configuration Explorer (ICE) is an *interactive visual analytics* tool that helps analyze and understand storage systems. It addresses the limitations of existing techniques, such as dealing with high-dimensional spaces and infusing domain knowledge, by making it easy for humans to understand and explore large parameter spaces. We described ICE and presented several exemplary case studies on a typical storage system to demonstrate how it can help analyze and understand performance efficiency, stability, etc. We believe that interactive visual analytics such as ICE, possibly in conjunction with other techniques (e.g., Parallel Coordinates [18] or Data Context Maps [11]), can greatly improve our ability to manage complex computer systems. ICE has the potential to pave the way for more applications of interactive visual analytics to storage research, leading to better understanding and more robust design of storage systems.

Acknowledgments

We would like to thank the anonymous USENIX HotStorage reviewers and our shepherd, George Amvrosiadis, for their valuable comments. We also thank Ibrahim "Umit" Akgun, Tyler Estro, and Amrith Arunachalam for the help with designing ICE. This work was made possible in part thanks to Dell-EMC, NetApp, and IBM support; NSF awards CNS-1251137, CNS-1302246, CNS-1305360, CNS-1622832, CNS-1650499, and CNS-1730726; and ONR award N00014-16-1-2264.

6 Discussion Topics

In this position paper, we propose to apply interactive visual analytics in storage research. We designed and prototyped ICE, which uses an enhanced box plot whose form is well understood by storage researchers, with an embedded density plot for throughput distribution, to present the data to the user in a compact, easily interpretable form. We expect our position paper to raise discussion issues on the storage research community from the following perspectives.

- What kind of visual approaches have the audience been using during their research on storage systems? How effective did they find their visual analytics approaches?
- How much does the audience believe interactive visual analytic approaches can benefit and help analyzing and understanding storage system behavior?
- What feedback would the audience provide to improve the design of ICE? How useful are the specific visualizations we designed? How well would they scale?
- Are there any other potential applications or uses cases for ICE in particular or other interactive visual analytic approaches in storage research?

Note: we plan to demo ICE interactively as part of any (extended) discussion of this position paper, and let others "play" with it as well.

References

[1] George Amvrosiadis and Vasily Tarasov. Filebench github repository, 2016. https://github.com/filebench/filebench/wiki.

[2] Wolfgang Barth. *Nagios: System and network monitoring*. No Starch Press, 2008.

[3] Jayanta Basak, Kushal Wadhwani, and Kaladhar Voruganti. Storage workload identification. *ACM Transactions on Storage*, 12(3):14:1–14:30, May 2016.

[4] Tobias Bleifuß, Leon Bornemann, Dmitri V. Kalashnikov, Felix Naumann, and Divesh Srivastava. DBChEx: Interactive exploration of data and schema change. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, Asilomar, CA, January 2019.

[5] Cynthia Brewer and Mark Harrower. Color Brewer 2.0. http://colorbrewer2.org/. Visited May 10, 2019.

[6] Cynthia A. Brewer, Geoffrey W. Hatchard, and Mark A. Harrow. ColorBrewer in print: A catalog of color schemes for maps. *Cartography and Geographic Information Science*, 30(1):5–32, 2003.

[7] Zhen Cao. *A Practical Auto-Tuning Framework for Storage Systems*. PhD thesis, Stony Brook University, January 2019.

[8] Zhen Cao, Vasily Tarasov, Hari Raman, Dean Hildebrand, and Erez Zadok. On the performance variation in modern storage stacks. In *Proceedings of the 15th USENIX Conference on File and Storage Technologies (FAST)*, pages 329–343, Santa Clara, CA, February-March 2017. USENIX Association.

[9] Zhen Cao, Vasily Tarasov, Sachin Tiwari, and Erez Zadok. Towards better understanding of black-box auto-tuning: A comparative analysis for storage systems. In *Proceedings of the Annual USENIX Technical Conference*, Boston, MA, July 2018. USENIX Association. Data set at http://download.filesystems.org/auto-tune/ATC-2018-auto-tune-data.sql.gz.

[10] Ming Chen, Dean Hildebrand, Henry Nelson, Jasmit Saluja, Ashok Subramony, and Erez Zadok. vNFS: Maximizing NFS performance with compounds and vectorized I/O. In *Proceedings of the 15th USENIX Conference on File and Storage Technologies (FAST)*, pages 301–314, Santa Clara, CA, February-March 2017. USENIX Association.

[11] Shenghui Cheng and Klaus Mueller. The data context map: Fusing data and attributes into a unified display. *IEEE transactions on visualization and computer graphics*, 22(1):121–130, 2016.

[12] Jeffrey Dean and Luiz André Barroso. The tail at scale. *Communications of the ACM*, 56(2):74–80, February 2013.

[13] Deborah Estrin, Mark Handley, John Heidemann, Steven McCanne, Ya Xu, and Haobo Yu. Network visualization with the VINT network animator nam. Technical Report 99-703b, University of Southern California, March 1999. http://www.isi.edu/~johnh/PAPERS/Estrin99d.pdf.

[14] Eduard Glatz, Stelios Mavromatidis, Bernhard Ager, and Xenofontas Dimitropoulos. Visualizing big network traffic data using frequent pattern mining and hypergraphs. *Computing*, 96(1):27–38, 2014.

[15] Mingzhe Hao, Gokul Soundararajan, Deepak Kenchammana-Hosekote, Andrew A. Chien, and Haryadi S. Gunawi. The tail at store: A revelation from millions of hours of disk and SSD deployments. In *14th USENIX Conference on File and Storage Technologies (FAST 16)*, pages 263–276, 2016.

[16] Jerry L. Hintze and Ray D. Nelson. Violin plots: A box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[17] Aapo Hyvärinen and Erkki Oja. Independent component analysis: algorithms and applications. *Neural networks*, 13(4-5):411–430, 2000.

[18] Alfred Inselberg and Bernard Dimsdale. Parallel coordinates. In *Human-Machine Interactive Systems*, pages 199–233. Springer, 1991.

[19] Nikolai Joukov, Ashivay Traeger, Rakesh Iyer, Charles P. Wright, and Erez Zadok. Operating system profiling via latency analysis. In *Proceedings of the 7th Symposium on Operating Systems Design and Implementation (OSDI 2006)*, pages 89–102, Seattle, WA, November 2006. ACM SIGOPS.

[20] Kimberly Keeton, Cipriano Santos, Dirk Beyer, Jeffrey Chase, and John Wilkes. Designing for disasters. In *Proceedings of the Third USENIX Conference on File and Storage Technologies (FAST 2004)*, pages 59–72, San Francisco, CA, March/April 2004.

[21] Robert Kosara, Fabian Bendix, and Helwig Hauser. Parallel sets: Interactive exploration and visual analysis of categorical data. *IEEE Transactions on Visualization & Computer Graphics*, 12(4):558–568, 2006.

[22] Tim Kraska, Alex Beutel, Ed H. Chi, Jeffrey Dean, and Neoklis Polyzotis. The case for learned index structures. In *Proceedings of the 2018 International Conference on Management of Data*, SIGMOD '18, pages 489–504, 2018.

[23] J. B. Kruskal. Multidimensional scaling by optimizing goodness of fit to a nonmetric hypothesis. *Psychometrika*, 29(1):1–27, 1964.

[24] Jundong Li, Kewei Cheng, Suhang Wang, Fred Morstatter, Robert P Trevino, Jiliang Tang, and Huan Liu. Feature selection: A data perspective. *ACM Computing Surveys (CSUR)*, 50(6):94, 2017.

[25] Patrick McDaniel, Kevin Butler, Stephen Mclaughlin, Radu Sion, Erez Zadok, and Marianne Winslett. Towards a secure and efficient system for end-to-end provenance. In *Proceedings of the Second USENIX workshop on the Theory and Practice of Provenance (TAPP '10)*, San Jose, CA, February 2010. USENIX Association.

[26] Justin Meza, Qiang Wu, Sanjev Kumar, and Onur Mutlu. A large-scale study of flash memory failures in the field. In *Proceedings of the 2015 ACM International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS 2015)*, pages 177–190, Portland, OR, June 2015. ACM.

[27] Iyswarya Narayanan, Di Wang, Myeongjae Jeon, Bikash Sharma, Laura Caulfield, Anand Sivasubramaniam, Ben Cutler, Jie Liu, Badriddine Khessib, and Kushagra Vaid. SSD failures in datacenters: What? when? and why? In *Proceedings of the Ninth ACM Israeli Experimental Systems Conference (SYSTOR '16)*, pages 7:1–7:11, Haifa, Israel, May 2016. ACM.

[28] Neal Patwari, Alfred O. Hero III, and Adam Pacholski. Manifold learning visualization of network traffic data. In *Proceedings of the 2005 ACM SIGCOMM workshop on Mining network data*, pages 191–19 6. ACM, 2005.

[29] Ohad Rodeh, Haim Helman, and David Chambliss. Visualizing block IO workloads. *ACM Transactions on Storage (TOS)*, 11(2):6, 2015.

[30] Bianca Schroeder, Raghav Lagisetty, and Arif Merchant. Flash reliability in production: The expected and the unexpected. In *Proceedings of the 14th USENIX Conference on File and Storage Technologies (FAST)*, pages 67–80, Santa Clara, CA, February 2016. USENIX Association.

[31] Priya Sehgal, Vasily Tarasov, and Erez Zadok. Evaluating performance and energy in file system server workloads. In *Proceedings of the USENIX Conference on File and Storage Technologies (FAST)*, pages 253–266, San Jose, CA, February 2010. USENIX Association.

[32] Jonathon Shlens. A tutorial on principal component analysis. *arXiv preprint arXiv:1404.1100*, 2014.

[33] Carson Sievert, Chris Parmer, Toby Hocking, Scott Chamberlain, Karthik Ram, Marianne Corvellec, and Pedro Despouy. plotly: Create interactive web graphics via 'plotly.js'. R package version 4.9.0, 2019. https://rdrr.io/cran/plotly/. Visited May 10, 2019.

[34] R. Spillane, R. Sears, C. Yalamanchili, S. Gaikwad, M. Chinni, and E. Zadok. Story Book: An efficient extensible provenance framework. In *Proceedings of the First USENIX workshop on the Theory and Practice of Provenance (TAPP '09)*, San Francisco, CA, February 2009. USENIX Association.

[35] Tableau Software. Tableau. https://www.tableau.com/. Visited May 10, 2019.

[36] Gary K.L. Tam, Vivek Kothari, and Min Chen. An analysis of machine- and human-analytics in classification. *IEEE Transactions on Visualization and Computer Graphics*, 2017.

[37] Wenbo Tao, Xiaoyu Liu, Çagatay Demiralp, Remco Chang, and Michael Stonebraker. Kyrix: Interactive visual data exploration at scale. In *Proceedings of the Conference on Innovative Data Systems Research (CIDR)*, 2019.

[38] Vasily Tarasov, Saumitra Bhanage, Erez Zadok, and Margo Seltzer. Benchmarking file system benchmarking: It *IS* rocket science. In *Proceedings of HotOS XIII:The 13th USENIX Workshop on Hot Topics in Operating Systems*, Napa, CA, May 2011.

[39] Vasily Tarasov, Erez Zadok, and Spencer Shepler. Filebench: A flexible framework for file system benchmarking. *;login: The USENIX Magazine*, 41(1):6–12, March 2016.

[40] Soon Tee Teoh, Kwan Liu Ma, S. Felix Wu, and Xiaoliang Zhao. Case study: Interactive visualization for Internet security. In *Proceedings of the conference on Visualization'02*, pages 505–508. IEEE Computer Society, 2002.

[41] Dana Van Aken, Andrew Pavlo, Geoffrey J. Gordon, and Bohan Zhang. Automatic database management system tuning through large-scale machine learning. In *Proceedings of the 2017 ACM International Conference on Management of Data*, SIGMOD '17, pages 1009–1024, 2017.

[42] Mengzhi Wang, Kinman Au, Anastassia Ailamaki, Anthony Brockwell, Christos Faloutsos, and Gregory R. Ganger. Storage device performance prediction with CART models. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '04/Performance '04, pages 412–413, New York, NY, USA, 2004. ACM.

[43] Wesley Willett, Jeffrey Heer, and Maneesh Agrawala. Scented widgets: Improving navigation cues with embedded visualizations. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1129–1136, 2007.

[44] Erez Zadok, Aashray Arora, Zhen Cao, Akhilesh Chaganti, Arvind Chaudhary, and Sonam Mandal. Parametric optimization of storage systems. In *HotStorage '15: Proceedings of the 7th USENIX Workshop on Hot Topics in Storage*, Santa Clara, CA, July 2015. USENIX, USENIX.