The Gap between Visualization Research and Visualization Software (VisGap) (2021)
C. Gillmann, M. Krone, G. Reina, T. Wischgoll (Editors)

The Gap between Visualization Research and Visualization Software in High-Performance Computing Center

Tommy Dang[1], Ngan Nguyen[1], Jon Hass[2], Jie Li[1], Yong Chen[1] and Alan Sill[1]

[1]Texas Tech University, USA
[2]Dell Inc., USA

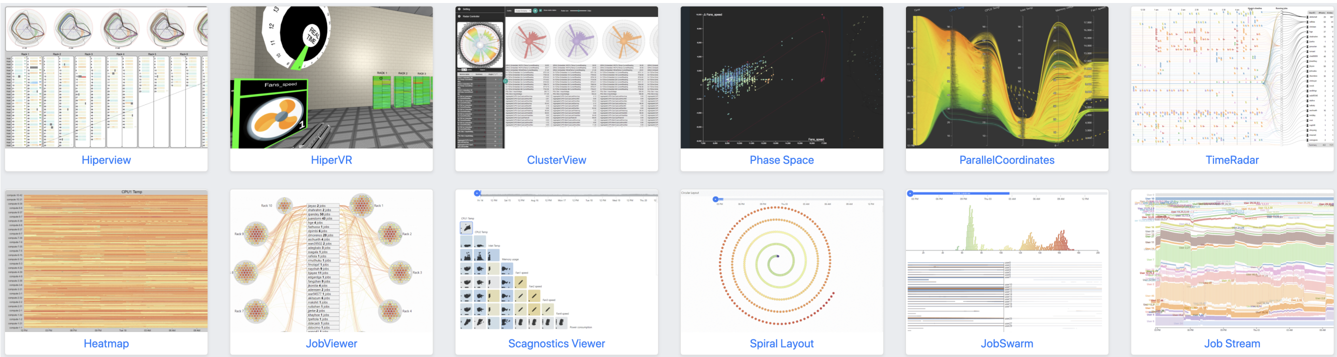

Figure 1: *Example visualization tools for visualizing and monitoring 467-node HPC cluster at Texas Tech University.*

Abstract

Visualizing and monitoring high-performance computing centers is a daunting task due to the systems' complex and dynamic nature. Moreover, different users may have different requirements and needs. For example, computer scientists carry out data analysis as batch jobs using various models, configurations, and parameters, and they often need to manage jobs. System administrators need to monitor and manage the system constantly. In this paper, we discuss the gap between visual monitoring research and practical applicability. We will start with the general requirements for managing high-performance computing centers and then share the experiences working with academic and industrial experts in this domain.

1. Introduction

Scientific experiments and discoveries rely on increasingly powered by advanced computing and data analysis capabilities [HTT*09]. We have witnessed exciting advancements in High-Performance Computing (HPC) hardware infrastructure capabilities over the past decades [Pal19]. Further advances in the scale of aggregated resources have taken place in the clouds [BSG*19]. However, the software infrastructure for HPC systems lags far behind the dramatic rate of hardware advancement [Sat20]. Participants in the HPC ecosystem desire to have software infrastructure to understand the status of HPC systems in a holistic manner, present valuable information in an intuitive way, and automate actions based on an integrated view. Unfortunately, such an integrated framework for a comprehensive view of HPC systems does not yet exist.

While it is not feasible to discuss the entire framework and its current issues in this paper, we will focus and the visualization research and tools for visualizing and monitoring HPC systems. Our project aims to provide visual approaches for situational awareness and health monitoring of HPC systems. The work is funded in part by the National Science Foundation through the IUCRC-CAC (Cloud and Autonomic Computing) Dell Inc. membership contribution. During this two-year project, we have opportunities to interact with both academic and industrial experts in HPC through the weekly meetings. We have built a set of visualizations to tackle different aspects/focuses of the HPC systems. However, not all of the generated visual representations are success stories. While some of the visualizations have been integrated into the Dell production system, others have never been used. This paper will highlight the common characteristics of successful adoption and mistakes in creating visualization software for the target users in this domain.

DOI: 10.2312/visgap.20211089

The rest of this paper is organized as follows: We summarize related visualization in Section 2 and then present requirement analysis in Section 3. We discuss the design and architecture of our visual interfaces in Section 4 and particularly illustrate the feasibility of the visualization tools on HPC use cases in Section 4.2. We also present further considerations and discussions in Section 4.3. Finally, the conclusion and future research direction are presented in Section 4.4.

2. Related Work

2.1. Monitoring tools for HPC systems

There are several open-source and commercial tools focusing on monitoring HPC systems. For instance, CARD [AP97], Parmon [Buy00] and Supermon [SM02] are early explorations of monitoring large-scale clusters. The Ganglia [MCC04] is a scalable distributed monitoring system that uses a multicast-based listen/announce protocol to monitor states within clusters. LDMS [AAB*14] allows various metrics such as memory usage, network bandwidth to be collected at a very high frequency. A few monitoring tools are capable of job-level monitoring. For example, Ovis [BDG*08] collects job data from the scheduler log file. Del Vento et al. [DVHE*11] proposed a method to associate floating-point counters with jobs and identifies poorly performing jobs. TACC Stats [EBB*14] runs the data collector module in the job scheduler prolog to collect job resource usage data. Other interesting monitoring tools built into frameworks that focus on cloud infrastructures include: OpenNebula [opea, opeb], CloudStack [clo], ZenPark [zen], Nimbus [nim], PCMONS [DCUW11], DARGOS [CFPMLS12], Hyperic-HQ [hyp], Sensu [sen], and a variety of projects from the Cloud Native Computing Foundation [cnc]. Many of these require extensive software infrastructure deployment and lack sufficient simplicity and flexibility to be generally applicable and easy to adopt in isolation without completely reformulating the infrastructure deployment.

2.2. Visualization tools for HPC systems

Visualization techniques and tools for monitoring and analyzing HPC systems also have a long history. LLView [KV15] is a client-server application that allows monitoring the utilization of clusters controlled by batch systems like IBM LoadLeveler, PBSpro, Torque, or IBM Blue Gene system database. However, multidimensional analysis is not possible in LLview. Nagios [Bar08] is a commonly used tool for HPC infrastructure monitoring, including hosts and associated hardware components, networks, storage, services, and applications. However, there are some issues with traditional Nagios, including human intervention requirements for the definition and maintenance of remote hosts configurations in Nagios Core, deployment of Nagios Remote Plugin Executor, and Nagios Service Check Acceptor (NSCA) on Nagios Server and each monitored remote host. Nagios also provides a basic web interface, but it can be a time-consuming task for system administrators to navigate through pages of reports on hosts, services, and status [AES05], and correlation in terms of temporal and spatial issues can be difficult [AAG03]. Amazon CloudWatch [Inc12] gives end users a web service to collect, view, and analyze pre-defined metrics in the form of logs, metrics, and events. Clients can define the thresholds for alarms, visualize logs/metrics beside each other, and take automated actions. Its primary disadvantage is that it has relatively few measurements and is only relevant to Amazon cloud assets. Splunk [Car12] is another software platform for mining and investigating log data for system analysts. Its most significant advantages are the capability to work with multiple data types in real-time [SCL10, ZK13], but can be slow when dealing with large amounts of visualization data [HG18]. Grafana [Gra19] provides an interactive visualization dashboard that enables users to view metrics via a set of widgets. Customized visualizations for analyzing high-dimensional data are, however, not supported.

3. Requirement analysis

The first step in producing the monitoring and visualization tools is to understand the user requirements [CGM*17]. Through in-depth discussions with HPC directors, systems administrators, and industry experts, we have identified prominent tasks important for monitoring HPC systems:

- **T1**: Provides spatial and temporal overview across hosts, racks, and other facilities over a given period of time,
- **T2**: Allows system administrators to filter by time-series features such as sudden changes in CPU temperatures for system troubleshooting,
- **T3**: Provide visual representations of how jobs run, behave, and progress over time.
- **T4**: Compare and contrast the resource usage (i.e.compute nodes, storage nodes, interconnection switches, power and cooling units, etc.) of different users and jobs in real-time.
- **T5**: Inspects the correlation of scheduling information vs. resource metrics via multidimensional analysis.
- **T6**: Integrates with the automation component so that the characterization and predictive analysis results of HPC systems using machine learning techniques can be visualized.

Besides the visualization tasks, we also identified the three main users of our visual frameworks.

- **U1 - System Administrators**: System administrators are concerned about the overall system performance. They need a comprehensive monitoring tool that connects jobs and resources in an integrated manner and a diagnosis tool to alert what happened in the system with the detailed job, resource, and computing metrics [SG14].
- **U2 - Domain Scientists**: Domain Scientists are concerned with the overall job performance. They need to keep track of their batch jobs in real-time via a user-friendly graphical interface and replay historical jobs to examine past behaviors [CHS*20].
- **U3 - Computer Scientists**: Different from System Administrators and Domain Scientists, Computer Scientists have specialized knowledge. They are concerned with tools for communication, threading, solvers, I/O, etc. typically make up the core foundation of the simulation [SHW*19, SAH*19].

Based on the requirements from the three primary user classes, we have developed prototypes of the framework components and have implemented preliminary tools for the use cases [met, Hipa, Hipb, Tim, Par, Hea, Job, Sca, Spi]. So far, all source code developed as a JavaScript-based web application using D3.js [BOH11] and is publicly available [iDV]. Several implementation methodologies and visualization tools developed by our group are already being adopted by an industry partner (Dell EMC) for integration with their production tools through closed-source development for licensed products.

4. Design and Architecture

Our project contains two main components: Data collection and data visualization.

4.1. Data Collection

The data collection component is the foundation of the framework and focuses on collecting data about jobs and resources using modern APIs and interfaces. The collected data will be organized and stored in databases optimized for speed and efficiency. The collected data will be populated to the visualization component through APIs and data frames, from which domain scientists, system administrators, and data scientists can access as they need via a graphical user interface. The data collection component comprises three main modules: a metrics collector module, a metrics storage module, and a metrics builder module. The *metrics collector* module provides the data collection service. This module retrieves data from a variety of sources, including the resource manager for jobs data and compute nodes, storage nodes, and other facilities for resources data via dedicated APIs. The *metrics storage* module provides data storage service. According to the data type for scalability and efficiency, this module stores collected data in pre-defined schemas into multiple databases, including time-series databases, NoSQL databases, and SQL databases. The *metrics builder* module provides the data query service. This module queries and accelerates data acquisition on requests and provides unified APIs and data frames for the automation and visualization components.

4.2. Data Visualization

An exploratory visual analysis involves both open-ended explorations of graphic patterns and concept-driven analysis when analysts have existing models or hypotheses [HVHC*08]. HPC scientists, administrators, and decision-makers mainly use the concept-driven approach, limiting the discovery of emerging issues and latent correlations in the highly dynamic and complex high-dimensional data generated by the HPC system. In this project, we develop and integrate both open-ended and concept-driven analysis in a unified visual framework that supports highlighting unusual correlations/causalities from a system overview, as well as detailed investigation on the events of interest [Shn97].

The visualization component is built on top of the data collection component to provide interactive visual representations for situational awareness and monitoring of HPC systems. The visualization requirements are expanded on the following dimensions: HPC spatial layout (physical location of resources in the system), temporal domain (as described in the metrics collector), and resource metrics (such as CPU temperature, fan speed, power consumption, etc.). As discuss in Section 3, the visualization also needs to incorporate the jobs scheduling and resource allocation data. We have developed a set of preliminary tools [ND19, Hipa, DNC21, Tim, Par, Hea, Job, Sca, Spi], over the course of 2 years in this project. In the next section, we will discuss each of this visualization w.r.t. the requirements/tasks identified in Section 3.

While it is too ambitious to provide the visual answers to all of these requirements, each of our preliminary tools tries to tackle different aspects of visualizing and analyzing the HPC systems. The following table provides a quick overview of our visualizations and the tasks that we could handle. Note that each of them is a separate tool. Each view (in the subsequent figures) was generated by a separate tool rather than loading the data once and generating three views by selecting a different analysis with a single integrated tool. The data used in the following examples are generated from the same 467-node cluster at Texas Tech University, even though the readings can be retrieved on different dates.

	T1	T2	T3	T4	T5	T6
HiperView [DNC21]	✓	✓				
SpacePhaser [DN19]		✓				✓
RadarViewer [NHC*20]	✓		✓		✓	
JobViewer [Job]	✓		✓	✓	✓	
ScagViewer [NNDH20]		✓			✓	
SpiralLayout [Spi]			✓	✓	✓	

Table 1: *Summary of our HPC visualization tools and their supporting tasks.*

4.2.1. HiperView

HiperView [DNC21] is a visual analytics prototype for monitoring and characterizing the health status of high-performance computing systems through a RESTful interface in real-time. *HiperView* aims to: 1) provide a graphical interface for tracking the health status of a large number of data center hosts in real-time statistics (visualization task **T1**), 2) to analyze unusual behavior of a series of events and to assist in performing preliminary troubleshooting and maintenance with a visual layout that reflects the actual physical locations (visualization task **T2**). Two use cases were analyzed in detail to assess the effectiveness of the *Visualization Tools* on a medium-scale, *Redfish*-enabled production high-performance computing system with a total of 10 racks and 467 hosts. The visualization apparatus has been proven to offer the necessary support for system automation and control. Our framework's visual components and interfaces are designed to potentially handle a larger-scale data center of thousands of hosts with hundreds of various health services per host. Figure 2 shows our real-time monitoring interface of the Quanah cluster (containing 467 hosts distributed in 10 racks from left to right) at the High-Performance Computing Center of Texas Tech University. Sudden changes, such as drops or increases in monitored parameters and flagging of outlying values, are visible on the bottom heatmap. Extremes in such readings can trigger administrative actions.

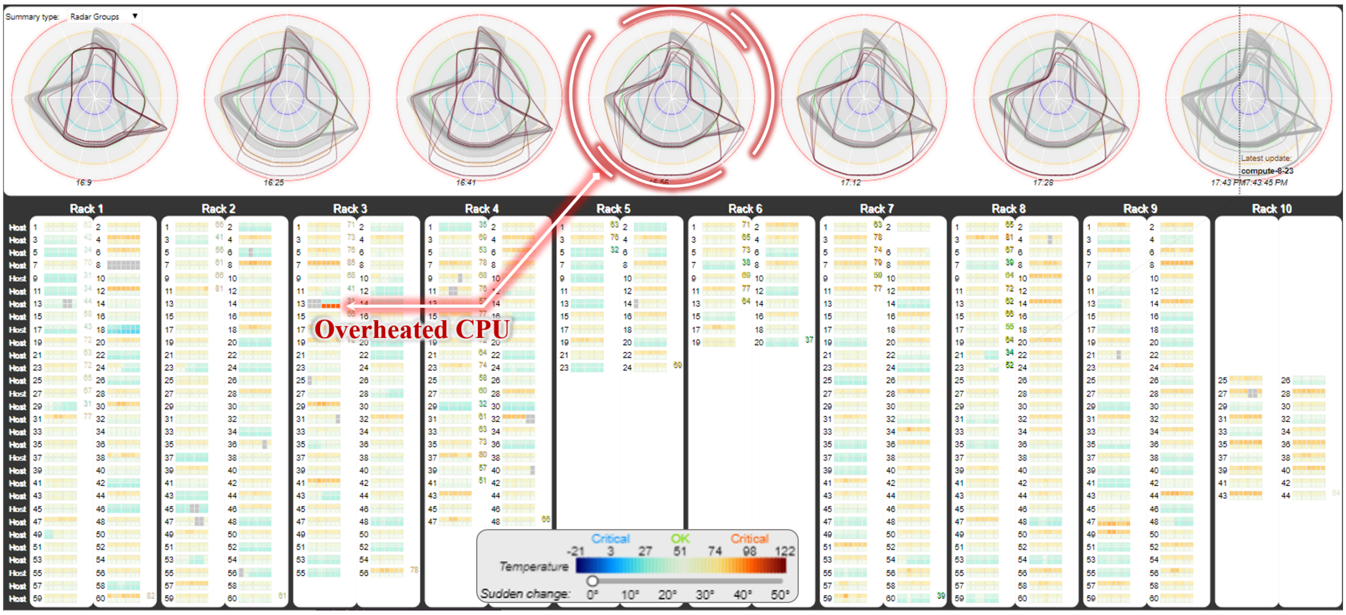

Figure 2: *Our HiperView interface for real-time monitoring 467 computing nodes of the Quanah cluster at Texas Tech University: red for high CPU temperature and blue for low CPU temperature.*

4.2.2. SpacePhaser

Inspired by the idea of *Phase space*, we apply this dynamical system theory [XZXS12] into dynamic behaviors of computing nodes in HPC systems. In particular, *SpacePhaser* [DN19] can capture abnormal dynamic correlations between various health dimensions in the data center (visualization task **T2**). Figure 3 illustrates *SpacePhaser* application to the health status of a data center at a university in one day: The CPU on *Compute-3-13* suddenly became overheated. In this use case, the color scale can be changed basing on distance from attractor or mean-squared error, measuring the average difference between the two temperature series. In detail, the abnormal behavior was alerted the system administrator to make CPU replacement for the malfunction CPU before it harms other neighboring CPUs. This abnormal event can be seen by the orange curve in Figure 3. The *Phase space* curves of computers are colored based on how far they are from the attractor (the average white curve in the middle): red for abnormal dynamics (strong variances) and blue for more stable computers. *SpacePhaser* is limited to viewing a selected reading at a time. This issue is resolved in the subsequent visualization.

4.2.3. RadarViewer

RadarViewer [NHC*20] visually summarizes the original temporal event sequences with clustering and, at the same time recovering individual sequences from the stacked radar chart. *RadarViewer* uses two strategies: *clustering* for grouping similar multivariate statuses into significant groups of interests using popular clustering methods [Har75]. Then, *superimposing* overlays multivariate representations on top of the clustering bundles. The challenge is to identify a set of clusters for a meaningful visual summary without imposing redundant patterns and inducing information loss [DPF16].

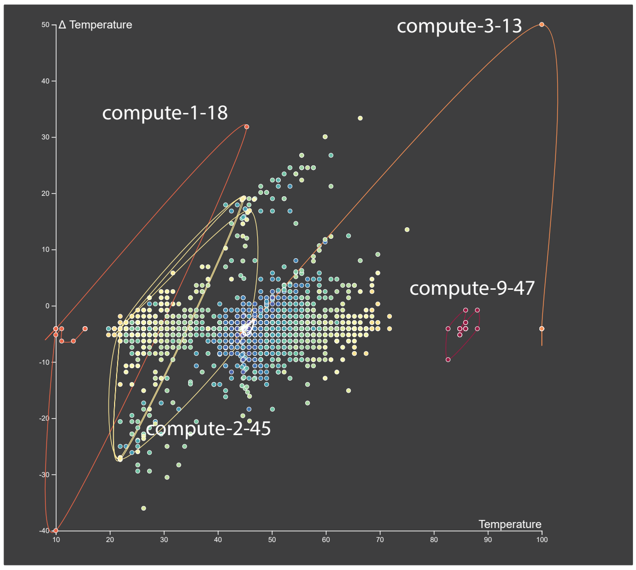

Figure 3: *Phase space visualization for the CPU temperatures of High-Performance Computing System: Four computing nodes (compute-1-18, compute-3-13, compute-2-45, and compute-9-47) with significant variances in CPU temperatures are highlighted in colored curves.*

To tackle this issue, we first define a small set of possible statuses within the multivariate data and then representing them onto the timeline where repeated statuses are simply compressed into a color-coded horizontal line.

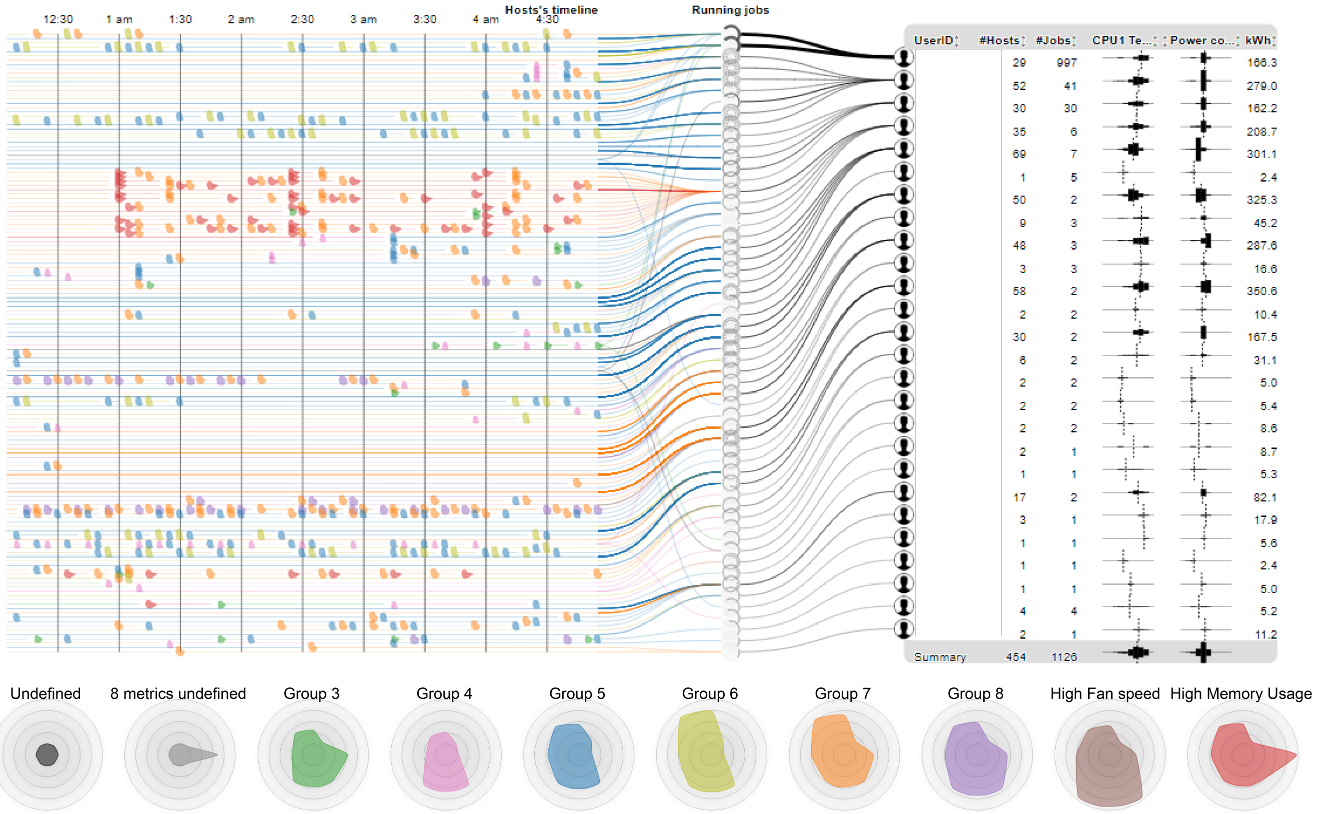

Figure 4: *Our RadarViewer interface for real-time monitoring health metrics and job scheduling: (left) Health metrics timeline which are projected into 10 major groups at the bottom (right) job scheduling information and user statistics such as **total energy usage** (kWh) at the last column.*

Figure 4 shows an example of *RadarViewer* for visualizing multivariate health metrics supplemented by job scheduling data [LAN*20]. The multivariate health statuses of computing hosts over an observed period are first clustered into ten major color-encoded groups(visualization task **T1**), as shown at the bottom of Figure 4. On the timeline view in the left panel, every line represents a computing node (or a group of computing nodes with similar visual signatures). The radar charts [MMP09] are superimposed on the associated timeline to present group switches (or major health status changes). The node timelines are connected to the corresponding active jobs and users table (visualization task **T3**), which provides statistics, such as average memory or total energy usage, as shown in the rightmost column (visualization task **T5**).

4.2.4. JobViewer

The strength of *JobViewer* [Job] is its ability to display both system health states and resource allocation information in a single view. It is easy to gain job allocation information in the main visualization. The clustering algorithm integrated into the application can quickly show the characteristics of the system health states. From these characteristics, we can point out any compute node with an irregular health state pattern to investigate the problems behind it. Moreover, *JobViewer* can allow us to observe the relations between jobs and compute node health (visualization task **T5**). This feature highlights jobs and users' behaviors to understand them better for improving or finding suspicious causes of any problem.

JobViewer provides administrators a powerful diagnosis tool. Figure 5 demonstrates an example of the Quanah cluster (containing 467 compute nodes distributed in 10 racks) at the High-Performance Computing Center (HPCC) of Texas Tech University (TTU) at 11:40 am on August 11, 2020. The middle list contains all current users of the system ordered by the number of requested jobs. The nodes in each rack are color-coded by the selected health metric (i.e., fan speed in this screenshot). We highlight several noticeable compute nodes in this example: black represents unreachable nodes (*node 8-12*), white represents unallocated nodes (*node 2-12*), black borders represent nodes shared by multiple users (*node 1-49*), and red represents nodes with high fan speed. By performing detailed investigation [AES05] of these compute nodes, a system administrator was able to track down *node 4-33* is the only node with high CPU temperature, while other neighboring nodes (*node 4-34*, *node 4-35*, and *node 4-36*) increased their fan speeds as they sensed the heat. *Node 4-33* was then replaced to prevent damages to the system. We can also set up an alert script where extreme health readings can trigger administrative actions automatically.

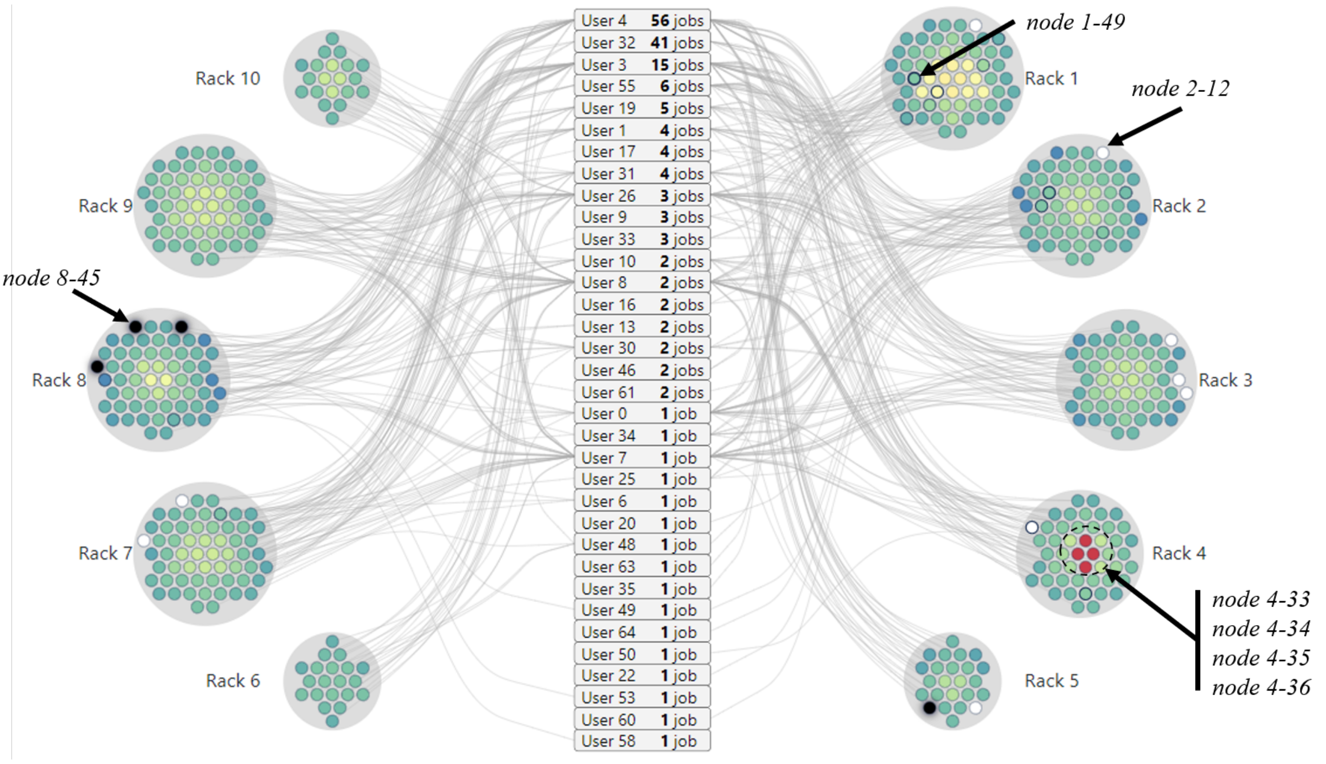

Figure 5: *Our JobViewer interface as one of real-time monitoring and diagnosis tools for 467-node Quanah cluster at Texas Tech University: red represents high fan speed and blue represents low fan speed.*

JobViewer also allows administrators to revisit the historical state and diagnose what happened in the system with detailed job and resource information (visualization task **T3**). *JobViewer* helps the interaction between administrators and users (domain scientists) as well for users' ticket or user support needs. For instance, *JobViewer* can visualize a resource under-utilization situation (e.g., a job only used one node out of ten requested) and explains the problem much more easily to users (often beginning users in such scenario) [WTN13, DK08]. The visualization tool is also useful to educate users so that they can detect issues in their jobs and improves the productivity of both scientists and administrators (visualization task **T4**).

4.2.5. ScagViewer

The Scagnostics are nine metrics for scoring different patterns of points distribution in a scatterplot, including Outlying, Skewed, Clumpy, Sparse, Striated, Convex, Skinny, Stringy, and Monotonic [WAG05]. *ScagViewer* [NNDH20] aims to extend the use of these Scagnostics to the high-dimensional time-series data sets, in which the temporal dimension is added into regular cross-sectional data, by using animation. We implement the approach on a web-based prototype with several visualization techniques to improve human recognition in the data exploration process based on the visual features of scatterplots [WAG06]. Our web-based prototype provides several features, as shown in Figure 6. The scatterplot matrix allows users to capture the whole data set at a particular

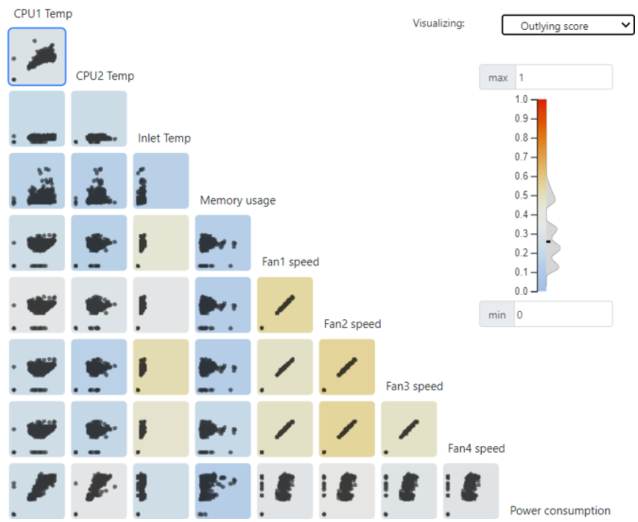

Figure 6: *The ScagViewer visualization for different computer metrics within the HPC system: Red is high Outlying plots while blue is low Outlying plots. Each dot in a scatterplot represents a computing node. Users can select a different visual feature (such as pairwise correlation or clusters) using the dropdown list on the top right corner.*

snapshot illustrated in the timeline. Each scatterplot in the matrix is colored according to its value of the chosen Scagnostics [DW14], such as Outlying or Monotonicity. The color scale is given along with the distribution of the chosen Scagnostics scores of all scatterplots in the matrix. Users can modify and filter the color scale by setting the thresholds on the selected visual feature (visualization task **T2**).

4.2.6. SpiralLayout

SpiralLayout [Spi] supports ranking users and jobs based on the resource usage and other metrics. In particular, we measure and rank the resource consumed by users/jobs/computing nodes over a certain period of time. Figure 7 shows an example of memory usage by four HPC users on one particular day. At each minute, we collect the memory readings of all computing nodes in the HPC system and rank them on a spiral layout where the outer rings are for high memory usage nodes, and the inner rings are for low consumption nodes. At the end of the day, we perform the heatmap visualizations based on the density of the computing nodes (on the spiral map) for each user (visualization task **T4**). As depicted in Figure 7, *User 25* consistently stay in the inner rings of our spiral map, and hence we can conclude that *User 25* consume the memory on this day. In contrast, *User 93* heatmap indicates that the computing nodes used by *User 93* mostly ranked as the top memory usages for the entire day. Therefore, system administration may want to investigate this case in order to avoid potential inappropriate usages of HPC systems.

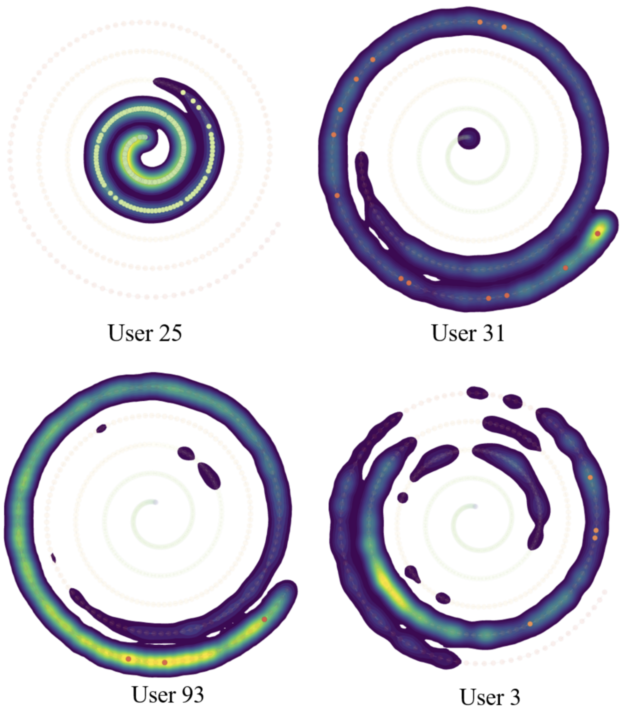

Figure 7: *The SpiralLayout visualization: Comparative user examples of memory usage in one day at the HPC Texas Tech University. The brighter areas are the most frequent positions of computing nodes by a given user.*

4.3. Discussions

As discussed, we have developed a good set of visualization tackling different dimensions of HPC systems. However, not all of the tools are useful for various classes of users. For example, the HPC director would like to learn the general issues and performances of the HPC system. The questions that he/she wants to answer could be straightforward: Are there any computing nodes overheat? Or Are there any users who seem suspicious? System administrators are more interested in detail investigation on an alerting event. Most of the time, more classical visual representations are more desirable since they do not require a lot of learning/adaption and visual reasonings.

System administrators are concerned about the machine domain composed of cores, computing nodes, racks, clusters, etc. In contrast, the domain scientist is concerned about the physical domain composed of nodes, cells, patches, etc. The common links are the tasks assigned to specific cores that operate on particular patches, which can be visualized in different ways. It the impractical to propose a visual solution that fits all needs of different users [SSH*19]. Instead, the customized views for each user are more flexible and reasonable [SBB*15, WLG*19].

4.4. Conclusions

This paper discusses the general requirements for managing a high-performance computing center, shares the experiences working with academic and industrial experts in this domain, and presents various case studies of visualization. As there are various needs from different users' classes, it is impractical to encapsulate all requirements of different HPC users into a single visualization. We have proposed various visual solutions during the course of two years and getting feedback through our weekly meetings. These visualizations tackle different aspects of HPC systems, such as computer health metrics, job scheduling, resources allocation, and their associations. In general, we would like to avoid the stiff learning curve and provide a holistic overview of the system before digging into details of an event of interest for system debugging.

Besides the main tasks and use cases, there are other subtasks and ongoing considerations that arise during the course of two years of this project. We will research these considerations and challenges in future work, including job scheduling and resource management visualization, customized and personalized visualization, and energy-awareness visualization.

5. Acknowledgments

The authors acknowledge the High-Performance Computing Center (HPCC) at Texas Tech University [hpc] in Lubbock for providing HPC resources and data that have contributed to the research results reported within this paper. The authors are thankful to the anonymous reviewers for their valuable feedback and suggestions that improved this paper significantly. This research is supported in part by the National Science Foundation under grant CNS-1362134, OAC-1835892, and through the IUCRC-CAC (Cloud and Autonomic Computing) Dell Inc. membership contribution.

References

[AAB*14] AGELASTOS A., ALLAN B., BRANDT J., CASSELLA P., ENOS J., FULLOP J., GENTILE A., MONK S., NAKSINEHABOON N., OGDEN J., ET AL.: The Lightweight Distributed Metric Service: A Scalable Infrastructure for Continuous Monitoring of Large Scale Computing Systems and Applications. In *SC'14: Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis* (2014), IEEE, pp. 154–165. 2

[AAG03] ANDRIENKO N., ANDRIENKO G., GATALSKY P.: Exploratory Spatio-Temporal Visualization: An Analytical Review. *Journal of Visual Languages & Computing 14*, 6 (2003), 503–541. 2

[AES05] AMAR R., EAGAN J., STASKO J.: Low-Level Components of Analytic Activity in Information Visualization. In *Proc. of the IEEE Symposium on Information Visualization* (2005), pp. 15–24. 2, 5

[AP97] ANDERSON E., PATTERSON D. A.: Extensible, Scalable Monitoring for Clusters of Computers. In *LISA* (1997), vol. 97, pp. 9–16. 2

[Bar08] BARTH W.: *Nagios: System and Network Monitoring*. No Starch Press, 2008. 2

[BDG*08] BRANDT J. M., DEBUSSCHERE B. J., GENTILE A. C., MAYO J. R., PÉBAY P. P., THOMPSON D., WONG M. H.: OVIS-2: A Robust Distributed Architecture for Scalable RAS. In *2008 IEEE International Symposium on Parallel and Distributed Processing* (2008), IEEE, pp. 1–8. 2

[BOH11] BOSTOCK M., OGIEVETSKY V., HEER J.: D3 Data-Driven Documents. *IEEE Trans. Vis. Comput. Graph. 17*, 12 (2011), 2301–2309. 3

[BSG*19] BELTRE A. M., SAHA P., GOVINDARAJU M., YOUNGE A., GRANT R. E.: Enabling hpc workloads on cloud infrastructure using kubernetes container orchestration mechanisms. In *2019 IEEE/ACM International Workshop on Containers and New Orchestration Paradigms for Isolated Environments in HPC (CANOPIE-HPC)* (2019), IEEE, pp. 11–20. 1

[Buy00] BUYYA R.: PARMON: a Portable and Scalable Monitoring System for Clusters. vol. 30, Wiley Online Library, pp. 723–739. 2

[Car12] CARASSO D.: *Exploring Splunk*. CITO Research New York, USA, 2012. 2

[CFPMLS12] CORRADI A., FOSCHINI L., POVEDANO-MOLINA J., LOPEZ-SOLER J. M.: DDS-Enabled Cloud Management Support for Fast Task Offloading. In *2012 IEEE Symposium on Computers and Communications (ISCC)* (2012), IEEE, pp. 000067–000074. 2

[CGM*17] CENEDA D., GSCHWANDTNER T., MAY T., MIKSCH S., SCHULZ H., STREIT M., TOMINSKI C.: Characterizing guidance in visual analytics. *IEEE Transactions on Visualization and Computer Graphics 23*, 1 (Jan 2017), 111–120. doi:10.1109/TVCG.2016.2598468. 2

[CHS*20] CHALKER A., HILLEGAS C. W., SILL A., BROUDE GEVA S., STEWART C. A.: Cloud and On-Premises Data Center Usage, Expenditures, and Approaches to Return on Investment: A Survey of Academic Research Computing Organizations. In *Practice and Experience in Advanced Research Computing*. 2020, pp. 26–33. 2

[clo] Apache CloudStack. http://cloudstack.apache.org/. URL: http://cloudstack.apache.org/. 2

[cnc] Cloud Native Computing Foundation. https://www.cncf.io/. URL: https://www.cncf.io/. 2

[DCUW11] DE CHAVES S. A., URIARTE R. B., WESTPHALL C. B.: Toward an Architecture for Monitoring Private Clouds. *IEEE Communications Magazine 49*, 12 (2011), 130–137. 2

[DK08] DAOUD M. I., KHARMA N.: A High Performance Algorithm for Static Task Scheduling in Heterogeneous Distributed Computing Systems. *Journal of Parallel and Distributed Computing 68*, 4 (2008), 399 – 409. URL: http://www.sciencedirect.com/science/article/pii/S0743731507000834, doi:https://doi.org/10.1016/j.jpdc.2007.05.015. 6

[DN19] DANG T., NGUYEN N.: Spacephaser: Phase space embedding visual analytics. In *2019 IEEE 43rd Annual Computer Software and Applications Conference (COMPSAC)* (2019), vol. 2, pp. 239–244. doi:10.1109/COMPSAC.2019.10213. 3, 4

[DNC21] DANG T., NGUYEN N., CHEN Y.: HiperView: real-time monitoring of dynamic behaviors of high-performance computing centers. *The Journal of Supercomputing* (2021). doi:https://doi.org/10.1007/s11227-021-03724-5. 3

[DPF16] DANG T. N., PENDAR N., FORBES A. G.: TimeArcs: Visualizing Fluctuations in Dynamic Networks. *Computer Graphics Forum* (2016). doi:10.1111/cgf.12882. 4

[DVHE*11] DEL VENTO D., HART D. L., ENGEL T., KELLY R., VALENT R., GHOSH S. S., LIU S.: System-Level Monitoring of Floating-Point Performance to Improve Effective System Utilization. In *SC'11: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis* (2011), IEEE, pp. 1–6. 2

[DW14] DANG T. N., WILKINSON L.: ScagExplorer: Exploring Scatterplots by Their Scagnostics. In *2014 IEEE Pacific Visualization Symposium* (March 2014), pp. 73–80. doi:10.1109/PacificVis.2014.42. 7

[EBB*14] EVANS T., BARTH W. L., BROWNE J. C., DELEON R. L., FURLANI T. R., GALLO S. M., JONES M. D., PATRA A. K.: Comprehensive Resource Use Monitoring for HPC Systems with TACC Stats. In *2014 First International Workshop on HPC User Support Tools* (2014), IEEE, pp. 13–21. 2

[Gra19] GRAFANA: The Open Platform for Beautiful Analytics and Monitoring, 2019. https://grafana.com/. 2

[Har75] HARTIGAN J.: *Clustering Algorithms*. John Wiley & Sons, New York, 1975. 4

[Hea] Heatmap. https://idatavisualizationlab.github.io/HPCC/heatmap/. 2, 3

[HG18] HUGH GREENBERG N. D.: Tivan: A Scalable Data Collection and Analytics Cluster. The 2nd Industry/University Joint International Workshop on Data Center Automation, Analytics, and Control (DAAC). 2

[Hipa] HiperJobViz: Visualizing Resource Allocations in High-Performance Computing Center via Multivariate Health-Status Data. https://idatavisualizationlab.github.io/HPCC/HiperJobViz/index.html. 2, 3

[Hipb] HiperView: Visualizing and Monitoring Health Status of High Performance Computing Systems. https://idatavisualizationlab.github.io/HPCC/HiperView/demo.html. 2

[hpc] High Performance Computing Center. http://www.depts.ttu.edu/hpcc/. URL: http://www.depts.ttu.edu/hpcc/. 7

[HTT*09] HEY T., TANSLEY S., TOLLE K. M., ET AL.: *The Fourth Paradigm: Data-Intensive Scientific Discovery*, vol. 1. Microsoft Research Redmond, WA, 2009. 1

[HVHC*08] HEER J., VAN HAM F., CARPENDALE S., WEAVER C., ISENBERG P.: Creation and Collaboration: Engaging New Audiences for Information Visualization. In *Information Visualization*. Springer, 2008, pp. 92–133. 3

[hyp] Hyperic Application & System Monitoring. https://sourceforge.net/projects/hyperic-hq/. URL: https://sourceforge.net/projects/hyperic-hq/. 2

[iDV] iDVL HPCC. https://github.com/iDataVisualizationLab/HPCC. 3

[Inc12] INC A.: Amazon CloudWatch, 2012. http://aws.amazon.com/cloudwatch/. 2

[Job] JobViewer. https://idatavisualizationlab.github.io/HPCC/jobviewer/index.html. 2, 3, 5

[KV15] KARBACH C., VALDER J.: System Monitoring: LLview. In *Computational Science and Mathematical Methods* (Nov 2015), Introduction to the programming and usage of the supercomputer resources at Jülich November 2015 (Course no. 78a/2015 in the training programme of Forschungszentrum Jülich), Jülich (Germany), 26 Nov 2015 - 27 Nov 2015. URL: https://juser.fz-juelich.de/record/279901. 2

[LAN*20] LI J., ALI G., NGUYEN N., HASS J., SILL A., DANG T., CHEN Y.: MonSTer: An Out-of-the-Box Monitoring Tool for High Performance Computing Systems. In *Cluster'20: Proceedings of IEEE International Conference on Cluster Computing* (2020), IEEE. 5

[MCC04] MASSIE M. L., CHUN B. N., CULLER D. E.: The Ganglia Distributed Monitoring System: Design, Implementation, and Experience. *Parallel Computing 30*, 7 (2004), 817–840. 2

[met] Metrics Builer API. https://influx.ttu.edu:8080/ui/. URL: https://influx.ttu.edu:8080/ui/. 2

[MMP09] MEYER M., MUNZNER T., PFISTER H.: MizBee: A Multiscale Synteny Browser. *IEEE Transactions on Visualization and Computer Graphics 15*, 6 (Nov. 2009), 897–904. URL: http://dx.doi.org/10.1109/TVCG.2009.167, doi:10.1109/TVCG.2009.167. 5

[ND19] NGUYEN N., DANG T.: HiperViz: Interactive Visualization of CPU Temperatures in High Performance Computing Centers. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (Learning)* (New York, NY, USA, 2019), PEARC '19, ACM, pp. 129:1–129:4. URL: http://doi.acm.org/10.1145/3332186.3337959, doi:10.1145/3332186.3337959. 3

[NHC*20] NGUYEN N., HASS J., CHEN Y., LI J., SILL A., DANG T.: Radarviewer : Visualizing the dynamics of multivariate data. In *Practice and Experience in Advanced Research Computing* (New York, NY, USA, 2020), PEARC '20, Association for Computing Machinery, p. 555–556. URL: https://doi.org/10.1145/3311790.3404538, doi:10.1145/3311790.3404538. 3, 4

[nim] Nimbus. http://www.nimbusproject.org/. URL: http://www.nimbusproject.org/. 2

[NNDH20] NGUYEN N. V. T., NGUYEN B. D. Q., DANG T., HASS J.: Scagnosticsviewer: Tracking time series patterns via scagnostics meatures. In *Proceedings of the 13th International Symposium on Visual Information Communication and Interaction* (New York, NY, USA, 2020), VINCI '20, Association for Computing Machinery. URL: https://doi.org/10.1145/3430036.3430072, doi:10.1145/3430036.3430072. 3, 6

[opea] OpenNebula. https://opennebula.io/. URL: https://opennebula.io/. 2

[opeb] OpenNebula Wikipedia. https://en.wikipedia.org/wiki/OpenNebula. URL: https://en.wikipedia.org/wiki/OpenNebula. 2

[Pal19] PALMER J.: More Super Supercomputers, 2019. 1

[Par] ParallelCoordinates. https://idatavisualizationlab.github.io/HPCC/ParallelCoordinates/index.html. 2, 3

[SAH*19] STEWART C. A., APON A., HANCOCK D. Y., FURLANI T., SILL A., WERNERT J., LIFKA D., BERENTE N., CHEATHAM T., SLAVIN S. D.: Assessment of Non-Financial Returns on Cyberinfrastructure: A Survey of Current Methods. In *Proceedings of the Humans in the Loop: Enabling and Facilitating Research on Cloud Computing*. 2019, pp. 1–10. 2

[Sat20] SATO M.: The Supercomputer Fugaku and Arm-SVE Enabled A64FX Processor for Energy-Efficiency and Sustained Application Performance. In *2020 19th International Symposium on Parallel and Distributed Computing (ISPDC)* (2020), IEEE, pp. 1–5. 1

[SBB*15] SCHULZ M., BHATELE A., BÖHME D., BREMER P.-T., GAMBLIN T., GIMENEZ A., ISAACS K.: *A Flexible Data Model to Support Multi-domain Performance Analysis*. 01 2015, pp. 211–229. doi:10.1007/978-3-319-16012-2_10. 7

[Sca] Scagnostics. https://idatavisualizationlab.github.io/HPCC/scagnosticsViewer/index.html. 2, 3

[SCL10] STEARLEY J., CORWELL S., LORD K.: Bridging the Gaps: Joining Information Sources with Splunk. In *SLAML* (2010). 2

[sen] Sensu. https://sensu.io/. URL: https://sensu.io/. 2

[SG14] SOMASUNDARAM T. S., GOVINDARAJAN K.: CLOUDRB: A Framework for Scheduling and Managing High-Performance Computing HPC Applications in Science Cloud. *Future Generation Computer Systems 34* (2014), 47 – 65. Special Section: Distributed Solutions for Ubiquitous Computing and Ambient Intelligence. URL: http://www.sciencedirect.com/science/article/pii/S0167739X13002884, doi:https://doi.org/10.1016/j.future.2013.12.024. 2

[Shn97] SHNEIDERMAN B.: *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, 3rd ed. Addison-Wesley Longman Publishing Co., Inc., USA, 1997. 3

[SHW*19] STEWART C. A., HANCOCK D. Y., WERNERT J., FURLANI T., LIFKA D., SILL A., BERENTE N., MCMULLEN D. F., CHEATHAM T., APON A., ET AL.: Assessment of Financial Returns on Investments in Cyberinfrastructure Facilities: A Survey of Current Methods. In *Proceedings of the Practice and Experience in Advanced Research Computing on Rise of the Machines (learning)*. 2019, pp. 1–8. 2

[SM02] SOTTILE M. J., MINNICH R. G.: Supermon: A High-Speed Cluster Monitoring System. In *Proceedings. IEEE International Conference on Cluster Computing* (2002), IEEE, pp. 39–46. 2

[Spi] Spiral Layout. https://idatavisualizationlab.github.io/HPCC/spiralLayout/index.html. 2, 3, 7

[SSH*19] SANDERSON A., SCHMIDT J., HUMPHREY A., PAPKA M., SISNEROS R.: In situ visualization of performance metrics in multiple domains. In *2019 IEEE/ACM International Workshop on Programming and Performance Visualization Tools (ProTools)* (2019), pp. 62–69. doi:10.1109/ProTools49597.2019.00014. 7

[Tim] TimeRadar. https://idatavisualizationlab.github.io/HPCC/TimeRadar/. 2, 3

[WAG05] WILKINSON L., ANAND A., GROSSMAN R.: Graph-Theoretic Scagnostics. In *Proceedings of the IEEE Information Visualization 2005* (2005), IEEE Computer Society Press, pp. 157–164. 6

[WAG06] WILKINSON L., ANAND A., GROSSMAN R.: High-dimensional visual analytics: Interactive exploration guided by pairwise views of point distributions. *IEEE Transactions on Visualization and Computer Graphics 12*, 6 (2006), 1363–1372. 6

[WLG*19] WOOD C., LARSEN M., GIMENEZ A., HUCK K., HARRISON C., GAMBLIN T., MALONY A.: *Projecting Performance Data over Simulation Geometry Using SOSflow and ALPINE*. 04 2019, pp. 201–218. doi:10.1007/978-3-030-17872-7_12. 7

[WTN13] WARRENDER R. L., TINDLE J., NELSON D.: Job Scheduling in a High Performance Computing Environment. In *2013 International Conference on High Performance Computing Simulation (HPCS)* (July 2013), pp. 592–598. doi:10.1109/HPCSim.2013.6641474. 6

[XZXS12] XIANG R., ZHANG J., XU X.-K., SMALL M.: Multiscale characterization of recurrence-based phase space networks constructed from time series. 013107. 4

[zen] ZenPack. https://github.com/zenoss/ZenPacks.zenoss.CloudStack. URL: https://github.com/zenoss/ZenPacks.zenoss.CloudStack. 2

[ZK13] ZADROZNY P., KODALI R.: *Big Data Analytics Using Splunk: Deriving Operational Intelligence from Social Media, Machine Data, Existing Data Warehouses, and Other Real-time Streaming Sources*. Apress, 2013. 2