VisGap'23 – The Gap between Visualization Research and Visualization Software (2023)
C. Gillmann, M. Krone, G. Reina, T. Wischgoll (Editors)

Reflections on the Developments of Visual Analytics Systems for the K Computer System Log Data

J. Nonaka[1], K. Fujita[1,2], T. Fujiwara[3], N. Sakamoto[2], K. Yamamoto[1], M. Terai[1], T. Tsukamoto[1], F. Shoji[1]

[1]RIKEN R-CCS, Operations and Computer Technologies Division, Japan
[2]Kobe University, Graduate School of System Informatics, Japan
[3]Linköping University, Department of Science and Technology, Sweden

Abstract

Flagship-class high-performance computing (HPC) systems, also known as supercomputers, are large, complex systems that require particular attention for continuous and long-term stable operations. The K computer was a Japanese flagship-class supercomputer ranked as the fastest supercomputer in the Top500 ranking when it first appeared. It was composed of more than eighty thousand compute nodes and consumed more than 12 MW when running the LINPACK benchmark for the Top500 submission. A combined power substation, with a natural gas co-generation system (CGS), was used for the power supply, and also a large air/water cooling facility was used to extract the massive heat generated from this HPC system. During the years of its regular operation, a large log dataset has been generated from the K computer system and its facility, and several visual analytics systems have been developed to better understand the K computer's behavior during the operation as well as the probable correlation of operational temperature with the critical hardware failures. In this paper, we will reflect on these visual analytics systems, mainly developed by graduate students, intended to be used by different types of end users on the HPC site. In addition, we will discuss the importance of collaborative development involving the end users, and also the importance of technical people in the middle for assisting in the deployment and possible continuation of the developed systems.

CCS Concepts

• ***Human-centered computing*** → ***Visualization systems and tools;*** *Visual analytics;* • ***Hardware*** → *Robustness;*

1. Introduction

Stable and uninterrupted operation is highly demanded for the operation of any HPC system. However, hardware failures can be considered inherent in the long-term operation of any HPC system due to the large number of hardware components involved. For instance, as shown in Fig. 1, the K computer system [MKS*12] was composed of 82,944 SPARC64 VIIIfx CPUs, 82,944 Interconnect Controllers (ICCs), 663,552 DRAM memory modules, and 10,368 power supply units. The K computer system consumed more than 12 MW of power during the LINPACK benchmark [Don88] when it surpassed the barrier of 10 PFlops for the first time in the Top 500 list [top]. To remove the massive heat generated during the regular operation, the cooling facility produced chilled water with a temperature around 15^oC for removing the heat from the CPUs and ICCs, and produced cooled air around 17^oC for removing the heat from the memory modules and power supply units inside the compute racks. Sensor information data sampled at different frequencies have been gathered and stored as big log datasets and used for analyzing from statistical perspectives, and we can cite the computational resource allocation [YUM*14], hardware failure analysis [Sho16], and energy efficiency [TSTY20].

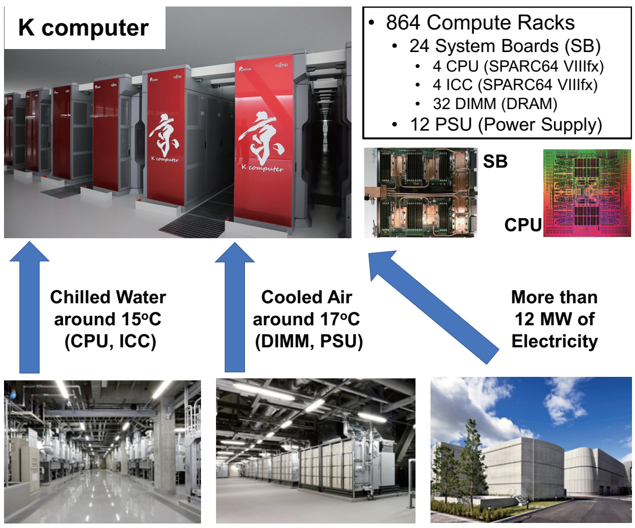

Figure 1: *Overview of the K computer system and its facility.*

DOI: 10.2312/visgap.20231116

With the advancements in sensor technology and measurement techniques, a variety of measurements, in the form of time-series log data, can be obtained from the HPC systems and infrastructure facilities. It is worth noting that there is currently increasing attention to Operational Data Analytics (ODA) [BJB*19], which is focused on the near real-time monitoring, archiving, and analysis of the HPC system and facility, as shown in the survey by Ott et al. [OSB*20]. Currently, the K computer was replaced by an even larger, power-hungry HPC system, the Fugaku supercomputer [SKTO22], which has been incorporating such ODA oriented mechanism, and a part of the data has been publicly available via Grafana-based dashboard, as shown in Fig. 2. However, in this work, we are mostly interested in the post-analysis of the stored log data, trying to obtain valuable information and knowledge from the data aiming to ameliorate the operation and maintenance of the HPC system and its facility.

Although the log data types, as well as the target goals, are different, we can identify various works addressing such a direction. For instance, log analyses of different supercomputers at Sandia National Laboratories and Lawrence Livermore National Laboratory [OS07] are examples of such post analyses. There are also GPU failure analyses for the Titan supercomputer at the Oak Ridge Leadership Computing Facility [RMJM19], and a more complete analysis (power, energy, and thermal dynamics), including GPU failures, for its successor, the Summit supercomputer [SOK*21]. We can also cite a visual analytics tool developed for identifying characteristic patterns from the HPC system behaviors and failures on the Theta Cray XC40 supercomputer at the Argonne Leadership Computing Facility (ALCF) [SLE*19]. At the ALCF, a cooling system analysis, including coolant monitor failures, was also conducted for the Mira supercomputer [RPK*21].

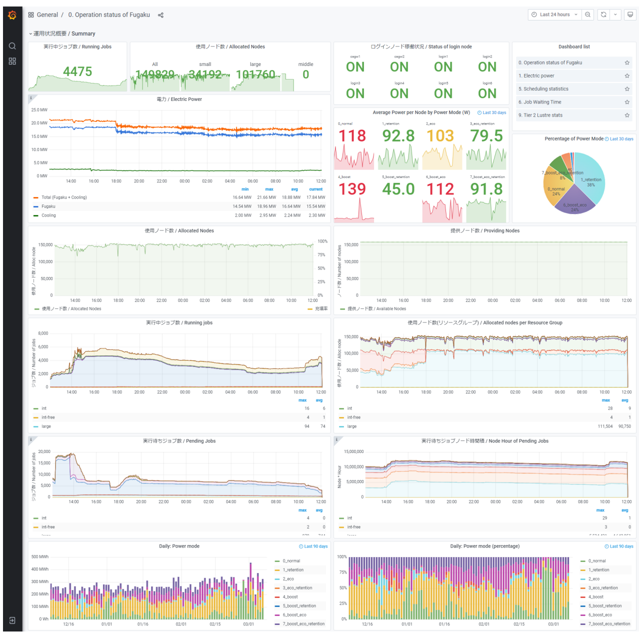

Figure 2: *Web-based dashboard for the monitoring and analysis of the operational status of the Fugaku supercomputer at RIKEN R-CCS. Publicly accessible from the following URL:* `https://status.fugaku.r-ccs.riken.jp/`.

In this work, we present several visual analytics systems that are collaboratively developed with academic partners and discuss observed gaps from the perspectives of the end user and third party. Our observation shows that, as in other application domains, the gaps due to mismatches of goals and expectations between the developer and user sides are easily caused in visualization research projects for HPC systems. The gaps we faced still remain challenging to be resolved and highlight the need for a multitude of considerations for future research projects, including the succession of projects with available student resources and the deployment process of research results for practical usage.

2. Log Data Analysis

There was a transition period, of more than a year and a half, from the end of the K computer operation in August 2019 until the start of the official operation of a new supercomputer, Fugaku in March 2021. During this period, an on-premise, OpenStack-based private cloud system (Fig. 3) was installed to assist the K computer users in their post-hoc analyses of simulation results. The Fugaku supercomputer debuted as the most powerful supercomputer in the Top500 ranking and is composed of 158,976 compute nodes

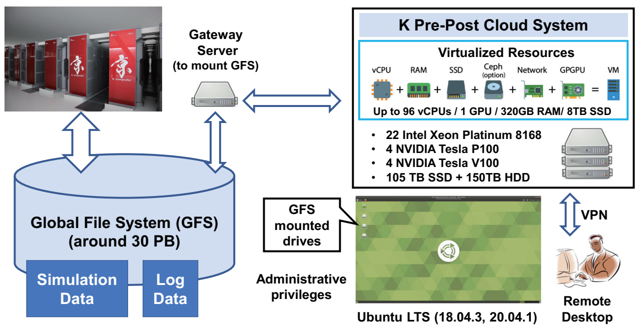

Figure 3: *OpenStack-based private cloud system (K Pre/Post Cloud) for assisting the post-hoc analyses of vast amounts of data stored on the K computer's persistent storage (Global File System).*

Table 1: *Example of CSV data from the K computer and its facility.*

Sampling rate	Contents
10 min.	I/O (among Node and Local File System) 82,944 nodes
5 min.	864 Compute racks 1,163 measurements (temperature, voltage, on/off position, etc.)
1 min.	Power consumption (per rack) 192 racks (kW)
1 min.	2 Co-Generation Systems 16 measurements (power generation, vapor generation, etc.)

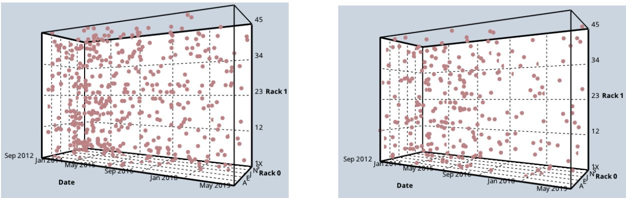

Figure 4: *Spatiotemporal distribution of critical DIMM (left) and CPU (right) failures requiring component substitution.*

(ARM-based Fujitsu A64FX CPUs) distributed over 432 compute racks (reduced to half compared to the 864 racks of the K computer) arranged in a planar space and connected via a high-speed network. It is worth mentioning that the power and cooling facility has been upgraded to deal with the increase in power consumption and heat generation. Considering the existence of a vast amount of log data gathered during the regular operation of the K computer, during the transition period, we focused on trying to better understand the operational behavior of such a large HPC system and its facility. Several measurements and information have been gathered at different sampling rates and stored as CSV-based log data. Table 1 shows some of them as illustrative examples. In addition, there was also a maintenance information sheet with a list of the replaced components due to the hardware failure. Fig. 4 shows a spatiotemporal distribution of the DIMM and CPU failures that required hardware replacement. Initially, simple 2D plots, obtained from gnuplot and Python-based plotting tools, have been used to analyze the aforementioned log datasets on the on-premise cloud system.

We began a research project by extending an existing collaboration with a research group at Kobe University to work with the above time-varying and multivariate log data. Later, we extended this collaboration with a research group at UC Davis (University of California, Davis) as they were also working on a similar topic. Some graduate students participated in this project, and incorporated the work in their dissertations, and we can say that it was a success from an academic perspective. In the developments, the target end users, with different computational skills, have participated in the evaluation and feedback. In addition, some technical staff has assisted with the deployment for practical usage. As in the statement of the scope of the VisGap Symposium, while most of the developed prototype systems have shown their value and potential, the end users are still waiting for further work to enhance them as practical tools for daily work. In the following section, we describe some of the visual analytics tools, mainly developed by graduate students, and discuss some lessons learned.

3. Visual Analytics Systems

The K computer enabled the users to run large batch-based jobs using tens of thousands of compute nodes for up to 24 hours, as well as huge batch jobs by using the entire K computer system for up to 8 hours during periodical large-scale job execution seasons. During its regular operation, many small and medium size jobs have co-existed; thus, critical hardware failures, which require the substitution of the failed components, may impact the jobs of the users. Due to its undesirable consequences, statistical hardware failure analyses have been carried out during the operational period [YUM*14, Sho16]. The Arrhenius model, shown in Eq. 1, describes the relationship between the lifetime of components and the temperature, which is aging [KK11, VdSdSM*19] in other words, and has been considered in the design of the K computer's CPU [Tak12].

$$L = Ae^{\frac{E_a}{kT}} \tag{1}$$

In this equation, L corresponds to the lifetime, A is a constant, E_a corresponds to the activation energy expressed in eV, k corresponds to the Boltzmann's constant ($8.6 \times 10^{-5} eV/K$), and T corresponds the temperature expressed in the kelvins. To investigate such a probable correlation between temperature and critical hardware failures, several visual analytics tools have been developed, as described in the next subsections.

3.1. Biclustering and Transfer Entropy-Based System

To understand the correlation between the temperature and critical hardware failures, it is essential to review a set of temperature data (CPU, intake/exhaust air to/from the racks, cooling water supply to the racks) measured during the regular operation of the K computer in conjunction with maintenance information regarding the failed component substitution. To facilitate the system development, data pre-processing has been carried out to reduce the size of the data since the CPU temperature was measured for the entire set (82,944 CPUs) every 5 minutes. The averaged CPU temperature per rack (total of 864 racks) and per day has been generated and used as the target data to be analyzed.

A graduate student from Kobe University, at that time, worked on

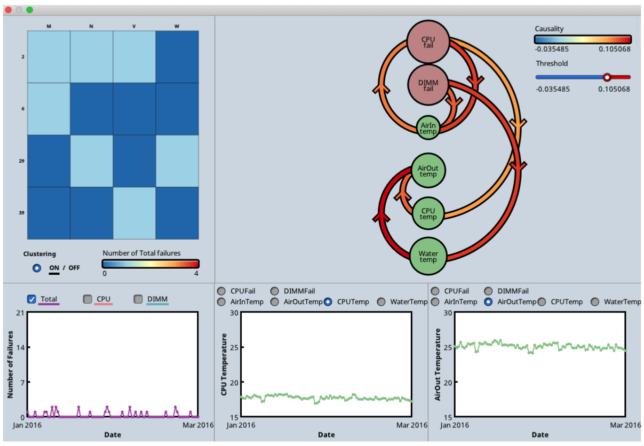

Figure 5: *Overview of the developed visual analytics system. The brushable data and failure plot screens (located on the lower part) are used to select a period of interest and the target failure for the analysis. The heatmap screen (upper left) shows the spectral biclustering result and the causal network screen (upper right) shows the causal relationship in the form of colored directed arcs.*

Proceedings published by Eurographics - The European Association for Computer Graphics.

a visual analytics system [KSNS18, KSN*19] for this matter. The GUI of the developed system is shown in Fig. 5. In this system, users can flexibly choose a period of interest by brushing the *data plot screen* based on the failure information shown in the *failure plot screen*. Then, the *heatmap screen* for the critical hardware failures, with its spatial locations, is updated, accordingly. The transfer entropy (TE) [Sch00] is used to estimate the causal index (strength of the impact of one on the other) from the resulting heatmap, and the obtained results are plotted in the form of circles and arcs on the *causal network screen* for visual causal relationship analysis. This system also uses spectral biclustering (SBC) [TSS05] to filter the input data (matrix data shown as the heatmap) for the TE calculation. The SBC is used to delimit the data space, that is, instead of utilizing the entire data space of the heatmap, the SBC will try to delimit the racks involved in the CPU failures as much as possible, and avoid involving unnecessary data in the TE calculation.

The causal network screen, which visualizes the causal indices as easy-to-understand directional graphs with color mapping, was highly rated by the end users during its practical evaluation. This visual representation is an amelioration from the initial prototype visualization using the thickness to represent the degree of causality. We should also mention that Fig. 4 was used as the *failure plot screen* for the initial prototype system. This gradual improvement is one of the positive consequences of collaborative development. Another lesson learned during this development is that it can lead to a better understanding of the visualization results by better knowing the pros and cons (or limitations) of the proposed approach, compared to the traditional use of the plotting tools as black boxes. Although the student was well motivated to work on further improvements by incorporating the convergent cross-mapping method, the lack of time for the development before his graduation (M. Sc.) and the lack of a candidate to take over the ongoing work required us to suspend the development. We hope that other well-motivated students will be able to accomplish the research before moving to a potential production run, with a focus on the log data of the Fugaku supercomputer (instead of the K computer), which is constantly collected over time after starting its operation.

3.2. Two-Step Dimensionality Reduction-Based System

The previous visual analytics system used averaged data, which can lead to overlooking relevant aspects during the analysis. Graduate students from UC Davis worked on a dimensionality reduction (DR) based visual analytics system capable of handling the original raw data without averaging them, which was named *MulTiDR* [FSS*21] (Fig. 6). This system was developed as a general-purpose framework to support the effective analysis of multivariate time-series data, which includes the log data from HPC systems and their facility. The MulTiDR employs two-step DR to generate an overview of the data and supports the interpretation of the DR results, using contrastive learning [FKM20] and interactive visualization. In the first step of DR, MulTiDR compresses and converts a third-order tensor into a matrix, and then, in the second step, it projects high-dimensional data points into a lower-dimensional space. Similar to the existing DR methods, the two-step DR result shows similarities of instances, variables, or time points, and enables visual identification of essential patterns, such as clusters and outliers.

Figure 6: *MulTiDR on the Ubuntu 20.04.1 LTS GNOME remote desktop, using the K Pre/Post Cloud system, during the analysis of a hardware failure day. User-selected clusters on the two-step DR view (upper left) are highlighted in the rack view (upper right) showing the physical coordinates of the racks with similar behaviors.*

Fig. 6 shows a visual analysis result of the entire log data from the K computer system for a single day. In addition to the temperature measurements, voltage measurements from the PSUs, and information from the air cooling fans installed on the compute racks comprising 1,163 different measures have been used. Since these measurements are collected every 5 minutes then 1,440 timestamps are generated in a single day. This single-day log data collected from 864 racks was represented as a $T \times N \times D$ tensor where $T = 1,440$, $N = 864$, and $D = 1,163$ with more than 1.4 billion elements in total. From the pair of scatterplots showing the two-step DR results, users can select clusters of interests that will be colored differently from each other. The racks belonging to the selected clusters will be also highlighted in the rack view (located on the top right), which shows the actual physical positions of the K computer racks. Additional line chart-based visualizations assist in better understanding the selected instances, valuables, or time points. For instance, the feature contribution visualization helps users characterize each of the selected clusters.

The MuTiDR has an easy-to-interact user interface for selecting clusters of interest and the web-based client implementation greatly facilitates the usage even for those who are accustomed to using Windows OS during their daily work. While MulTiDR is a general-purpose system,

the GUI showed in Fig. 6 was tailored for the K computer's log data analysis. In addition, the command line-based data conversion tool was also tailored for the K computer's log data. The offline data conversion and the MulTiDR server-side setup on the K Pre/Post Cloud system could be conducted without difficulty by people accustomed to the Linux environment. Although it is a well-known fact, we learned that it can be difficult to convince someone to use a different OS environment, and it becomes highly important to provide an application capable of running on the OS of the target end user. We also learned that the client/server implementation could greatly facilitate mitigating the problem by doing separately the data conversion and server setup for the end users.

Regarding the visual analysis, although it was easily possible to visually identify different clusters and their spatial distribution, via the rack view, the interpretation of the variables was still difficult due to the large number of corresponding entities (1,163 entities). Although the developer implemented an extension on the data conversion tool to improve the selection of the variables for facilitating this interpretation, we learned that this type of customization for each practical use would be considered a non-research activity, and it might be difficult to ask graduate students for such efforts.

It is worth mentioning that this additional modification gave important clues to the user side for further customization such as the conversion of natural gas co-generation system (CGS) log data [Sek12] and the turbostat log data from a different supercomputer (JCAHPC Oakforest-PACS) [NHS20]. The availability of the source code as well as minimum necessary documentation was also crucial to build and use on different hardware systems. For instance, Fig. 7 shows the MulTiDR running on the current Fugaku Pre/Post environment.

3.3. Functional Data Analysis-Based System

The graduate students from UC Davis also developed a visual analytics system [SFS*22] incorporating functional data analysis (FDA) [RS05] to analyze consecutively measured data. The selected log data for the analysis was composed of temperature and voltage measurements, where abnormal values have the potential to induce malfunctions and failures. It is worth mentioning that continuously updating data inherently has infinite dimensions, and FDA methods often suffer from high computational costs when dealing with large time series data. To analyze the log data in a streaming manner, a progressive algorithm for instantaneously generating the magnitude-shape (MS) plot [DG17] was developed, In addition, this system also enables the augmentation of analysis using the MS plot with functional principal component analysis (FPCA) and interactive visualizations to aid in reviewing clusters identified from the MS plot.

The original MS plot requires recomputation when new time points and/or time series are added (e.g., adding temperatures obtained from different compute racks). To provide useful intermediate results or to enable the incremental addition of time points and time series, the developed progressive algorithm generates the MS plot with estimated directional outlyingness measures in addition to a refinement mechanism to maintain the MS plot quality. A performance evaluation was carried out using filtered data composed of 390 temperature measurements per compute rack (336,960 measurements in total) by using a MacBook Pro (13-inch, 2019) with a 2.8 GHz Quad-Core Intel Core i7 processor and 16 GB 2,133 MHz LPDDR3 memory. Considering the existence of a 2-week data with 4,032 time points (collected every 5-minute interval), the update of the MS plot without the progressive algorithm (i.e., a recalculation

Figure 7: *Running the MS plot algorithm (left side) and MulTiDR (right side) on the RHEL8 Xfce remote desktop using the Fugaku Pre/Post Environment (via Open OnDemand). The MulTiDR loaded the Oakforest-PACS supercomputer's turbostat log data, and is used for the identification and characterization of CPUs with similar temporal behaviors during high load conditions.*

on 4,033 time points) required 8.2 minutes, in contrast to the 2.5 seconds when applying the progressive algorithm. For the exclusion and addition of time series (with the size of $4,032 \times 336,959$), the subsequent update of the MS plot with approximation was completed in 16.6 seconds, in comparison to the overall update time of 8.4 minutes without using the progressive algorithm.

The prototype visual analytics framework, which implemented the aforementioned algorithm, also provided an easy-to-interact user interface to select desired outliers to be analyzed. The *space view* facilitates easy identification of racks with a large number of outliers, and the *FPCA view* enables further analysis of the outliers by means of FPCA. The value and potential of the proposed approach were highly evaluated by visualization research peers and resulted in one of the three most highly rated papers from the 2022 IEEE Pacific Visualization Symposium (PacificVis) [CRZ22]. Although it can definitely be considered a success story from the academic perspective, we are still seeking a better way to apply the framework to practical analyses for the end users. At least, for now, we confirmed that the progressive MS plot algorithm is ready to be used on the current Fugaku Pre/Post environment, as shown in Fig. 7.

3.4. Sequential DR-Based System

A graduate student from Kobe University has worked on developing a visual analytics system [FSF*22] (Fig. 8) using DR and the same averaged data used in Sec. 3.1 to identify periods of time and spatial locations, where we see characteristic operational behaviors. It is worth noting that, in this development, the person in charge of the operation of the power and cooling facility has actively participated in the evaluation and feedback due to the interest in obtaining some knowledge from the previous facility (designed for the K computer). Supercomputers are shared computational resources and usually operate with different computational workloads at different locations (space) and timings (time). Therefore,

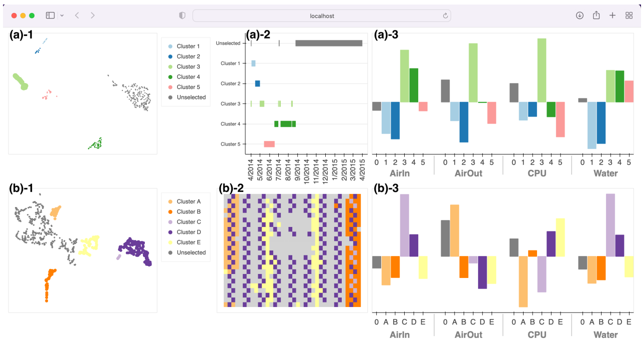

Figure 8: *Overview of the sequential DR-based visual analytics system. The upper part shows the DR plot (left) for temporal points and the temporal information plot of clusters. The lower part shows the DR plot (left) for spatial points and the spatial information plot of clusters.*

a better understanding of the supercomputer system's heat generation and cooling behavior was highly desired from the facility's operational side for decision-making and optimization planning.

The log data is represented as a third-order tensor (or 3D array) with the axes of time, space, and measured values and applies multiple DR steps based on a user-selected temporal or spatial region of interest corresponding to a certain time period or compute rack positions. While these multiple DR steps follow a similar procedure to the two-step DR utilized in MulTiDR (Sec. 3.2), by actively involving users to specify time points to be used in the second step, the system can mitigate the information loss caused by DR of the massive amount of elements. Through multiple DR steps, it produces a 2D scatterplot where users can find groups (clusters) of points (with spatial or temporal information) with similar, or depending on the point of view, dissimilar behaviors. PCA was applied to compress the data and Uniform Manifold Approximation and Projection (UMAP) [MHM18], which is a nonlinear DR method with low computational overhead, was applied to find similar points.

The web-based GUI of the implemented prototype system is shown in the Fig. 8, where the upper part represents the *temporal selection view* consisting of three plots: (a)-1 shows the DR results for temporal points; (a)-2 shows the temporal information; and (a)-3 shows each cluster's average of measurement values normalized to have unit variance. The lower part represents the *spatial selection view* consisting of three plots similar to the temporal selection view. From the scatterplot in (a)-1, users can select a temporal cluster of interest. Once a temporal cluster is selected, the temporal information of the selected cluster is shown on (a)-2, and the average measurement values for the selected clusters are plotted as a bar chart on (a)-3. Users can also visualize related information of the points, via the tooltip, by positioning the mouse cursor on the desired points. This interactive exploration process is repeated until the temporal clusters of interest are determined.

During the continuous development and improvement of the system, the developer has sometimes acted as the end user and made a variety of visual analyses to discuss with the operational staff, and this interaction was important not only for the validation of the developed system but also for the end user side to better understand the potential and limitations of the proposed approach. The graduate student was highly motivated and working on further enhancements, including functionality for enabling a more detailed comparative analysis. However, due to the graduation (M. Sc.), this work remains suspended similar to another work presented in Sec. 3.1. This issue is probably even peculiar to Japanese society, where most graduate students prefer to start working for a company instead of pursuing a doctoral course, resulting in a short period of time for dedicating to research and development. In addition, it is usually difficult to obtain sufficient support from former graduate students, and, therefore, it becomes highly important to obtain as much information, for instance, in the form of source codes and documentation, before their graduation.

4. Lessons Learned

Several people have participated and contributed in different ways to this set of projects at different periods, including some from the beginning. It was started as an unofficial collaborative research project involving universities (Kobe University and UC Davis) and a research institute (RIKEN R-CCS), where a win-win situation with mutual gains was expected. From the development side, access to the data and the list of needs were the requirements for initiating the research and development activities, and on the other hand, the user side was eager to try new tools not only for obtaining knowledge from the vast amounts of stored data but also for learning state-of-the-art techniques and approaches. Although it could be a well-known fact, we could reconfirm the importance of mutual understanding that the goals and expectations from both sides can be quite different. For instance, we can list some of them, as shown in Table 2.

It is also well-known that graduate students are usually under constant pressure for publications, and their research and development schedule usually follows the submission deadlines for academic events and journal publications. As a result, developed software for research purposes usually has only the necessary functionality for the evaluations required for the publications, and many of them are released as open source software as is. For the user side, it was highly important to understand this workflow, and not to expect production-level software from the graduate students' side. For the daily work usage, usually, an effective and stable software system with a detailed user manual and support is desired, and sometimes there is no issue to pay for the necessary software and service. Therefore, there is also a possibility to continue the developments

Table 2: *Goals and expectations from both sides.*

Development Side	User Side
Supervisors	Supervisors
Graduate students	R&D and Technical staffs
Publication	Practical use
Research purposes	Daily work usage
Evaluation quality	Manual and User support
Open Source Software (OSS)	Commercial software or OSS

such as customizations and functionality enhancements on the user side. However, for this purpose, we felt that a more close collaborative development from the beginning stage will be highly beneficial since it becomes difficult to undertake the development without knowing the details.

Focusing on the daily work usage, enhancements on the application's robustness can probably be outsourced considering the existing best programming practices and techniques. Performance and scalability are other points that require careful attention due to the volume of data and available hardware for data analysis. Since even a laptop could be used as the development platform by using small test case data; then, there is a possibility that the developed application may not be taking full advantage of the available resources on the target hardware system (multicore CPU, GPU, and abundant memory). We also observed that web-based applications could greatly facilitate the usage by different types of users, and in a similar manner, we see great potential in the web-based Open On-Demand [HJN*16] for providing an easy-to-use environment for the deployment and usage of visual data analysis applications, as shown in Fig. 7. This functionality was recently implemented on the Fugaku environment [NMY23]. We will continue exploring its potential for visual data analysis.

Similar to what we observed with the supercomputer users [NS20], we have different groups of users with different skills. On one side, we have highly skilled users who can customize or enhance by themselves, but at the same time, we have users who was accustomed to using Windows-based GUI applications. For the latter, the support of technical staff in the middle for the deployment and data pre-processing was crucial. We also learned that such people in the middle can play an important role in assisting the actual end users to understand the value and potential of the newly developed tools, and to convince the managerial level for obtaining the necessary budget for converting them to daily work software tools. In the end, one of the main lessons learned was the importance of clearly understanding the limits (e.g. available time, human resources, and others) of both sides and not pushing over the boundaries to have a harmonious, successful win-win situation.

5. Conclusions

In this paper, we made reflections on the visual analytics systems developed for analyzing log data generated during the years of operation of the K computer, a former Japanese flagship supercomputer. Several graduate students participated in this project proposing different approaches for the visual analysis of such large time-varying multivariate data. From the academic point of view, we can say without hesitation that it was a complete success with some papers being published in top-ranked journals including an honorable mention award. From the practical point of view, it would probably be better to have a further continuation in the development of each of the developed systems. However, it is worth mentioning that we observed several positive aspects of these developments as discussed in the lesson learned section. We are aware that each research and development program has its own peculiarity and the cases presented here might be too specific. However, we hope that our experience would be interesting for others who might face similar situations.

Acknowledgements

We are grateful to the former and current students at Kobe University and UC Davis who participated in this project. In this context, we are extremely grateful to Dr. Kwan-Liu Ma for enabling a fruitful collaboration involving his students. This work was partially supported by JSPS KAKENHI (Grant Number 20H04194, 21H04903) and the Knut and Alice Wallenberg Foundation (grant KAW 2019.0024).

References

[BJB*19] BOURASSA N., JOHNSON W., BROUGHTON J., CARTER D. M., JOY S., VITTI R., SETO P.: Operational Data Analytics: Optimizing the national energy research scientific computing center cooling systems. In *Proc. ICPPW* (2019), ACM. doi:10.1145/3339186.3339210. 2

[CRZ22] CAO N., ROPINSKI T., ZHAO J.: Guest editors' introduction: Special section on IEEE PacificVis 2022. *IEEE Transactions on Visualization and Computer Graphics 28*, 6 (2022), 2299–2300. doi:10.1109/TVCG.2022.3165393. 5

[DG17] DAI W., GENTON M.: Multivariate functional data visualization and outlier detection. *Journal of Computational and Graphical Statistics 27* (3 2017). doi:10.1080/10618600.2018.1473781. 5

[Don88] DONGARRA J. J.: The LINPACK benchmark: An explanation. In *Supercomputing* (1988), Houstis E. N., Papatheodorou T. S., Polychronopoulos C. D., (Eds.), Springer, pp. 456–474. 1

[FKM20] FUJIWARA T., KWON O.-H., MA K.-L.: Supporting analysis of dimensionality reduction results with contrastive learning. *IEEE Transactions on Visualization and Computer Graphics 26*, 1 (2020), 45–55. doi:10.1109/TVCG.2019.2934251. 4

[FSF*22] FUJITA K., SAKAMOTO N., FUJIWARA T., TSUKAMOTO T., NONAKA J.: A visual analytics method for time-series log data using multiple dimensionality reduction. *Journal of Advanced Simulation in Science and Engineering 9*, 2 (2022), 206–219. doi:10.15748/jasse.9.206. 5

[FSS*21] FUJIWARA T., SHILPIKA, SAKAMOTO N., NONAKA J., YAMAMOTO K., MA K.: A visual analytics framework for reviewing multivariate time-series data with dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics 27*, 2 (2021), 1601–1611. doi:10.1109/TVCG.2020.3028889. 4

[HJN*16] HUDAK D. E., JOHNSON D., NICKLAS J., FRANZ E., MCMICHAEL B., GOHAR B.: Open OnDemand: Transforming computational science through omnidisciplinary software cyberinfrastructure. In *Proc. XSEDE* (New York, NY, USA, 2016), ACM. doi:10.1145/2949550.2949644. 7

[KK11] KEANE J., KIM C. H.: An odometer for CPUs. *IEEE Spectrum 48*, 5 (2011), 28–33. 3

[KSN*19] KOISO K., SAKAMOTO N., NONAKA J., SHOJI F., YAMAMOTO K.: Biclustering and transfer entropy for the visual analysis of critical hardware failures on the K computer. Poster presented at IEEE PacificVis 2019, Bangkok, Thailand, 2019. 4

[KSNS18] KOISO K., SAKAMOTO N., NONAKA J., SHOJI F.: A transfer entropy based visual analytics system for identifying causality of critical hardware failures case study: CPU failures in the K Computer. In *Proc. AsiaSim* (2018), pp. 563–574. doi:10.1007/978-981-13-2853-4_44. 4

[MHM18] MCINNES L., HEALY J., MELVILLE J.: UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv:1802.03426* (2018). 6

[MKS*12] MIYAZAKI H., KUSANO Y., SHINJOU N., SHOJI F., YOKOKAWA M., WATANABE T.: Overview of the K computer system. *Fujitsu Scientific and Technical Journal 48*, 3 (2012), 255–265. 1

[NHS20] NONAKA J., HANAWA T., SHOJI F.: Analysis of cooling water temperature impact on computing performance and energy consumption. In *Proc. CLUSTER* (2020), pp. 169–175. doi:10.1109/CLUSTER49012.2020.00027. 5

[NMY23] NAKAO M., MIURA S., YAMAMOTO K.: Introduction of Open OnDemand to supercomputer Fugaku. Poster of HPC Asia, 2023. URL: https://www.sc-asia.org/wp-content/uploads/2023/02/post101.pdf. 7

[NS20] NONAKA J., SAKAMOTO N.: Lessons learned from large data visualization software development for the K computer. In *Proc. VisGap* (2020), Gillmann C., Krone M., Reina G., Wischgoll T., (Eds.), The Eurographics Association. doi:10.2312/visgap.20201113. 7

[OS07] OLINER A., STEARLEY J.: What supercomputers say: A study of five system logs. In *Proc. DSN* (2007), pp. 575–584. doi:10.1109/DSN.2007.103. 2

[OSB*20] OTT M., SHIN W., BOURASSA N., WILDE T., CEBALLOS S., ROMANUS M., BATES N.: Global experiences with HPC operational data measurement, collection and analysis. In *Proc. CLUSTER* (2020), pp. 499–508. doi:10.1109/CLUSTER49012.2020.00071. 2

[RMJM19] ROJAS E., MENESES E., JONES T., MAXWELL D.: Analyzing a five-year failure record of a leadership-class supercomputer. In *Proc. SBAC-PAD* (2019), pp. 196–203. doi:10.1109/SBAC-PAD.2019.00040. 2

[RPK*21] ROY R. B., PATEL T., KETTIMUTHU R., ALLCOCK W., RICH P., SCOVEL A., TIWARI D.: Operating liquid-cooled large-scale systems: Long-term monitoring, reliability analysis, and efficiency measures. In *Proc. HPCA* (2021), pp. 881–893. doi:10.1109/HPCA51647.2021.00078. 2

[RS05] RAMSAY J. O., SILVERMAN B. W.: *Functional Data Analysis*. Springer-Verlag New York, 2005, p. 428. 5

[Sch00] SCHREIBER T.: Measuring information transfer. *Physical Review Letters 85*, 2 (2000), 461. 4

[Sek12] SEKIGUCHI Y.: Construction and facilities technologies for the K computer. *Fujitsu Scientific and Technical Journal 48*, 3 (2012), 266–273. 5

[SFS*22] SHILPIKA, FUJIWARA T., SAKAMOTO N., NONAKA J., MA K.: A visual analytics approach for hardware system monitoring with streaming functional data analysis. *IEEE Transactions on Visualization and Computer Graphics 28*, 06 (2022), 2338–2349. doi:10.1109/TVCG.2022.3165348. 5

[Sho16] SHOJI F.: The K computer and its failures. Proc. FTXS, Association for Computing Machinery, p. 17. doi:10.1145/2909428.2909436. 1, 3

[SKTO22] SATO M., KODAMA Y., TSUJI M., ODAJIMA T.: Co-design and system for the supercomputer "Fugaku". *IEEE Micro 42*, 2 (2022), 26–34. doi:10.1109/MM.2021.3136882. 2

[SLE*19] SHILPIKA F., LUSCH B., EMANI M., VISHWANATH V., E. PAPKA M., MA K. L.: MELA: A visual analytics tool for studying multifidelity HPC system logs. In *Proc. DAAC* (2019), pp. 13–18. doi:10.1109/DAAC49578.2019.00008. 2

[SOK*21] SHIN W., OLES V., KARIMI A. M., ELLIS J. A., WANG F.: Revealing power, energy and thermal dynamics of a 200PF pre-exascale supercomputer. In *Proc. SC* (2021). 2

[Tak12] TAKAYAMA K.: Challenges to develop highly reliable HPC systems. Talk at the 1st JST International Symposium on Dependable VLSI Systems 2012, 2012. URL: https://www.jst.go.jp/crest/dvlsi/topics/12smpj/12DVLSISMPtakayama.pdf. 3

[top] Top500. https://www.top500.org. Accessed on 2023.3.11. 1

[TSS05] TANAY A., SHARAN R., SHAMIR R.: Biclustering algorithms: A survey. *Handbook of Computational Molecular Biology 9*, 1-20 (2005), 122–124. 4

[TSTY20] TERAI M., SHOJI F., TSUKAMOTO T., YAMOCHI Y.: A study of operational impact on power usage effectiveness using facility metrics and server operation logs in the K computer. In *Proc. CLUSTER* (2020), pp. 509–513. doi:10.1109/CLUSTER49012.2020.00072. 1

[VdSdSM*19] VIEIRA A. N. C., DE SOUZA P. S. S., DOS SANTOS MARQUES W., DA SILVA CONTERATO M., FERRETO T. C., LUIZELLI M. C., LORENZON A. F., FILHO A. C. S. B., ROSSI F. D., NONAKA J.: The impact of parallel programming interfaces on the aging of a multicore embedded processor. In *Proc. ISCAS* (2019), pp. 1–5. doi:10.1109/ISCAS.2019.8702116. 3

[YUM*14] YAMAMOTO K., UNO A., MURAI H., TSUKAMOTO T., SHOJI F., MATSUI S., SEKIZAWA R., SUEYASU F., UCHIYAMA H., OKAMOTO M., OHGUSHI N., TAKASHINA K., WAKABAYASHI D., TAGUCHI Y., YOKOKAWA M.: The K computer operations: Experiences and statistics. *Procedia Computer Science 29* (2014), 576–585. doi:https://doi.org/10.1016/j.procs.2014.05.052. 1, 3