RadarViewer : Visualizing the dynamics of multivariate data

Ngan Nguyen
Texas Tech University
Lubbock, Texas, USA
ngan.v.t.nguyen@ttu.edu

Jon Hass
Dell, Inc.
Austin, Texas, USA
jon.hass@dell.com

Yong Chen
Texas Tech University
Lubbock, Texas, USA
yong.chen@ttu.edu

Jie Li
Texas Tech University
Lubbock, Texas, USA
jie.li@ttu.edu

Alan Sill
Texas Tech University
Lubbock, Texas, USA
alan.sill@ttu.edu

Tommy Dang
Texas Tech University
Lubbock, Texas, USA
tommy.dang@ttu.edu

ABSTRACT

This showcase presents a visual approach based on clustering and superimposing to construct a high-level overview of sequential event data while balancing the amount of information and the cardinality in it. We also implement an interactive prototype, called *RadarViewer* , that allows domain analysts to simultaneously analyze sequence clustering, extract useful distribution patterns, drill multiple levels-of-detail to accelerate the analysis. The *RadarViewer* is demonstrated through case studies with real-world temporal datasets of different sizes.

KEYWORDS

Radar chart, time-series visualization, multivariate data analysis

ACM Reference Format:

Ngan Nguyen, Jon Hass, Yong Chen, Jie Li, Alan Sill, and Tommy Dang. 2020. *RadarViewer* : Visualizing the dynamics of multivariate data. In *Practice and Experience in Advanced Research Computing (PEARC '20), July 26–30, 2020, Portland, OR, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3311790.3404538

1 INTRODUCTION

We propose a visual approach based on two strategies: *clustering* for grouping similar multivariate statuses into significant groups of interests using popular clustering methods [2]. Then, *superimposing* overlays multivariate representations on top of the clustering bundles. Our technique, called *RadarViewer* , visually summarizes the original temporal event sequences with clustering and, at the same time recovering individual sequences from the stacked radar chart. The challenge is to identify a set of clusters for a meaningful visual summary without imposing redundant patterns and inducing information loss [1]. 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.

PEARC '20, July 26–30, 2020, Portland, OR, USA

ACM ISBN 978-1-4503-6689-2/20/07…$15.00
https://doi.org/10.1145/3311790.3404538

2 RELATED WORK

2.1 Event Sequence Visualization

A simple approach is to align the ordered events along the horizontal axis. For example, Lifelines [7] allows users to show multiple facets of the personal histories where each facet was displayed as an individual timeline. This visualization method can be considered as an extension of the Gantt chart since it imposed colors on facets to highlight the relationships among event sequences. However, the proposed method was not able to show trends. The authors solved this issue in the extended version, Lifelines2 [8], by aggregating multiple event sequences into a stacked bar chart over a time period. Still, this method required more space as the number of facets increased. To tackle this issue, CloudLines [4] compresses multiple timelines together to form a representation of a timeline bundle. This method is based on Kernel density estimation and truncation function. EventFlow [6], LifeFlow [9], and CoreFlow [5] used tree-like branching structure to abstract temporal event sequences.

3 *RADARVIEWER* DESIGN

We demonstrate our proposed visualization tool on monitoring the HPC health metrics. First, we adapt the existing clustering algorithms, such as k-means [3] to summarize the major statuses (or clusters) of multivariate data. Figure 1 where users can identify patterns, outliers, or clustering of interest. Each clustering radar chart represents one dimension of the event sequence data; the user can rename the dimension for easier navigation in the main interface by clicking on the icon at the top left of the radar chart.

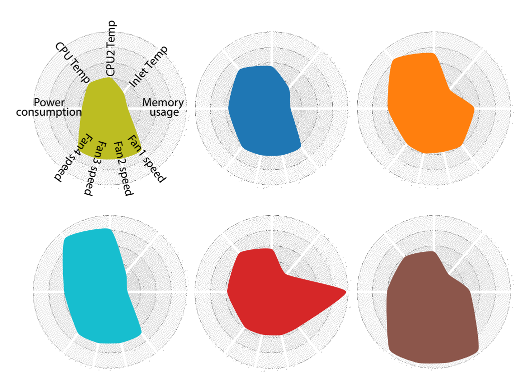

Figure 1: Major clusters generated by the k-means.

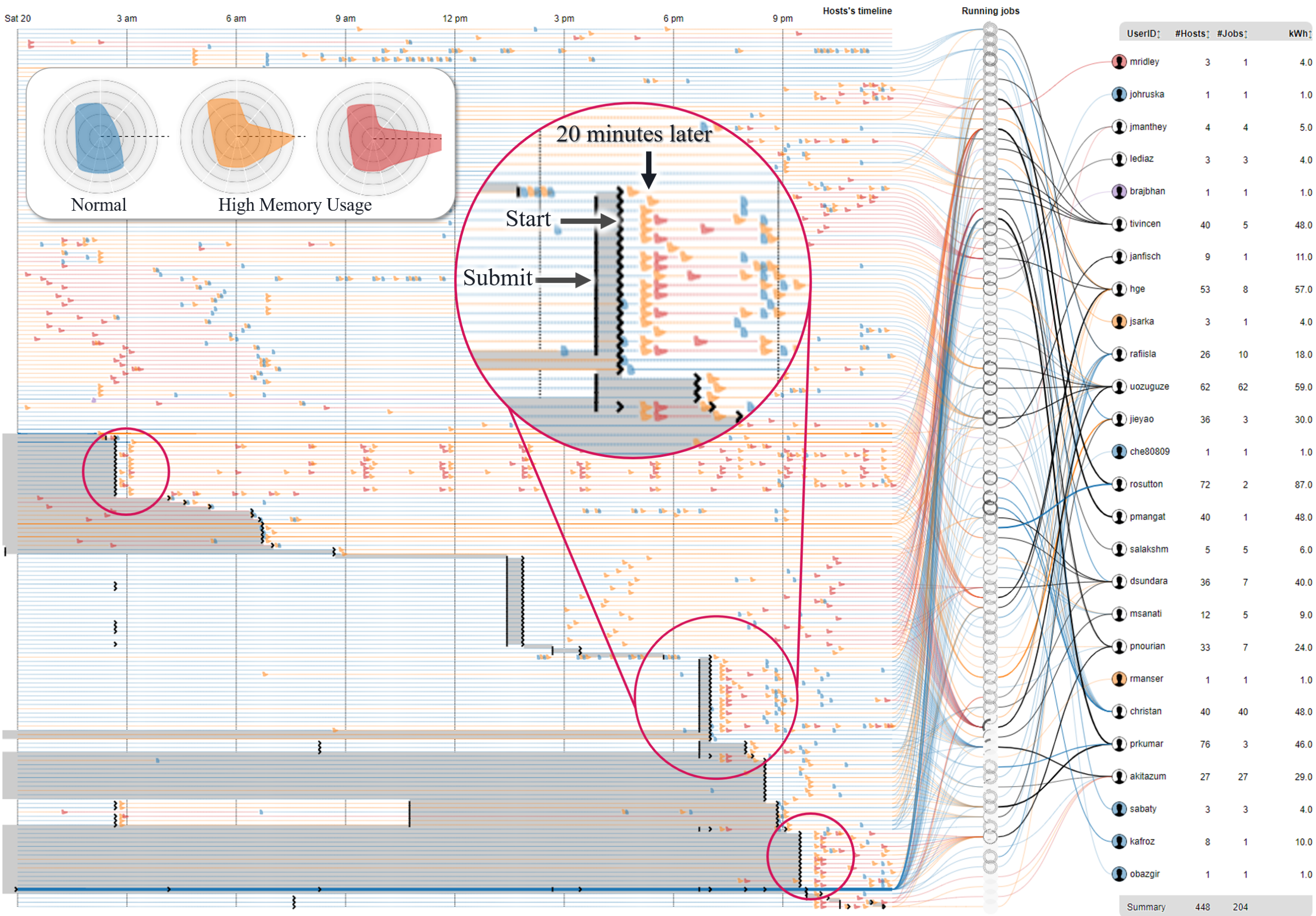

Figure 2: *RadarViewer* on HPC dataset: Red circles indicates enclose the time point when many instances change their status. The circles highlight the common patterns of high memory usage when the jobs are started on these computing nodes.

Figure 2 illustrates a use case where analysts can use *RadarViewer* to capture dynamic behaviors of multiple events (host) over time. Through the overview, the presence of many vertically aligned radar charts (with clustering and ordering) shows that there are abnormal activities around 3 am, 7 pm, and 10 pm where the system changes from normal activity to high memory usage. This information can be used for further investigation, such as which job causes the issues and who owns the corresponding jobs. Knowing this information in advance can help the system administrator to have better strategies for resource allocations and management.

4 CONCLUSION

This project presented a technique for visualizing temporal event sequences. Our proposed approach was based on the clustering and superimposing techniques to construct a high-level overview of sequential event data. *RadarViewer* allows domain analysts to simultaneously capture sequence clustering, extract useful distribution patterns, drill multiple levels-of-detail to accelerate interactive data analysis. The *RadarViewer* was demonstrated through case studies with real-world temporal datasets.

REFERENCES

[1] Tuan Nhon Dang, Nick Pendar, and Angus Graeme Forbes. 2016. TimeArcs: Visualizing fluctuations in dynamic networks. In *Computer Graphics Forum*, Vol. 35. Wiley Online Library, 61–69.

[2] J.A. Hartigan. 1975. *Clustering Algorithms.* John Wiley & Sons, New York.

[3] John A Hartigan. 1975. Clustering Algorithms. (1975).

[4] M. Krstajic, E. Bertini, and D. Keim. 2011. CloudLines: Compact Display of Event Episodes in Multiple Time-Series. *IEEE Transactions on Visualization and Computer Graphics* 17, 12 (Dec 2011), 2432–2439. https://doi.org/10.1109/TVCG.2011.179

[5] Zhicheng Liu, Bernard Kerr, Mira Dontcheva, Justin Grover, Matthew Hoffman, and Alan Wilson. 2017. Coreflow: Extracting and visualizing branching patterns from event sequences. In *Computer Graphics Forum*, Vol. 36. Wiley Online Library, 527–538.

[6] Megan Monroe, Rongjian Lan, Hanseung Lee, Catherine Plaisant, and Ben Shneiderman. 2013. Temporal event sequence simplification. *IEEE transactions on visualization and computer graphics* 19, 12 (2013), 2227–2236.

[7] Catherine Plaisant, Brett Milash, Anne Rose, Seth Widoff, and Ben Shneiderman. 1995. *LifeLines: visualizing personal histories.* Technical Report.

[8] T. D. Wang, C. Plaisant, B. Shneiderman, N. Spring, D. Roseman, G. Marchand, V. Mukherjee, and M. Smith. 2009. Temporal Summaries: Supporting Temporal Categorical Searching, Aggregation and Comparison. *IEEE Transactions on Visualization and Computer Graphics* 15, 6 (Nov 2009), 1049–1056. https://doi.org/10.1109/TVCG.2009.187

[9] Krist Wongsuphasawat, John Alexis Guerra Gómez, Catherine Plaisant, Taowei David Wang, Meirav Taieb-Maimon, and Ben Shneiderman. 2011. LifeFlow: visualizing an overview of event sequences. In *Proceedings of the SIGCHI conference on human factors in computing systems.* ACM, 1747–1756.