



## JobNet: 2D and 3D Visualization for Temporal and Structural Association in High-Performance Computing System

---

Ngan V.T. Nguyen, Huyen N. Nguyen, Jon Hass and Tommy Dang

EasyChair preprints are intended for rapid dissemination of research results and are integrated with the rest of EasyChair.

October 29, 2021

# *JobNet*: 2D and 3D Visualization for Temporal and Structural Association in High-Performance Computing System

Ngan V. T. Nguyen<sup>1</sup>[0000-0001-5039-8019], Huyen N. Nguyen<sup>1</sup>[0000-0001-6554-2327], Jon Hass<sup>2</sup>, and Tommy Dang<sup>1</sup>[0000-0001-8322-0014]

<sup>1</sup> Texas Tech University, Lubbock, Texas, United States

<sup>2</sup> Dell Inc., Austin, Texas, United States

**Abstract.** Performance monitoring in a High-Performance Computing (HPC) system is an essential and challenging task. With a large number of system components, coupled with health metrics that need to be reported, visualizing the system’s internal structure over time will uncover patterns and enable insights, empowering analysis from monitoring. This paper presents a visualization tool that visualizes the temporal and structural association of HPC system components using a force-directed graph layout algorithm. The visualization contains 2D and 3D representation, supporting a complete analysis of the compute usage, how users and job submission are interconnected throughout the observational interval. Design alternatives for time representation are discussed and depicted in 2D and 3D visualization encodings, with animation and exclusive presentation. The interaction capabilities of the tool assist visual exploration of health metrics and changes in system status over time. The tool’s usefulness and effectiveness in the monitoring task are demonstrated by a case study on a real-world HPC dataset.

**Keywords:** HPC monitoring · Force-directed layout · 2D and 3D visualization

## 1 Introduction

High-Performance Computing (HPC) supercharges innovative technologies to empower complex computation and deliver data insights with quick and accurate results. The innovations in the HPC system encompass the advancement in system components and simultaneously increases the complexity, making the monitoring task of such elements more demanding and challenging. It is important for the system administrators to inspect the system components and their structural association with their corresponding elements for the purpose of monitoring and analysis. For example, to examine the performance of a job running on a compute server, there needs to be put in the context of who is the user submitting that job, the status of health metric indicators at the time, what

kind of job that node is running, whether it is data-intensive, memory-intensive or other usage behavior category.

Graph visualizations are commonly used to present the complex relationship structure between internal components inside a system. One of the most well-known visualization techniques for visualizing node-link-based structure is force-directed layout [4]. The principle behind force-directed approaches lies in node positioning so that all edges are of approximately equal length and maintain minimum edge crossings as possible [7]. The equilibrium state is achieved by considering forces within the layout simulated as a physical system such that the edges are springs and the nodes are electrically charged particles.

Our objective is to show how interactive visualization can demonstrate the dynamic characteristic of temporal and structural association within the HPC system. We propose a new visualization metaphor in 2D and 3D for monitoring HPC system components. Our method is based on force-directed layout, applied for both the 2D and 3D views, which allows the representation of multiple servers, jobs, and users in the same view. In the 2D visualization, the animation is incorporated to foster an interactive interface, showing the temporal shifts in server usages. The new visualization metaphor in the 3D interface depicts the evolution over time for the entire observation period, accompanied by efficient use of screen space without cluttering the view. Finally, we validate this approach and demonstrate its usefulness by conducting a case study on a real-world HPC monitoring dataset. The contribution of this work can be laid out as follows:

- An interactive visualization for monitoring the temporal and structural association of HPC system components, called *JobNet*, which:
  - employs force-directed layout for visualizing multiple compute servers, jobs, and users in the same view
  - provides a 2D visualization showing dynamic characteristics of the network made of servers over time
  - presents a 3D representation of the entire observation period with emphasis on the temporal evolution
- Demonstration of the approach by applying it to a dataset acquired from a real-world HPC center. The tool is also applied to a movie dataset to examine its applicability.

The rest of this paper is organized as follows: Section 2 summarizes existing work that is close to our paper. Section 3 describes the domain problem leading to design rationale and decisions. Section 4 shows the findings through case studies with the real HPC data for the 2D and 3D visualization, along with a movie dataset. Section 5 concludes our paper with outlook for future work direction.

## 2 Related work

In HPC monitoring, visualization has presented as an intuitive approach for the representation of vast volumes of data, seeking to analyze and explore it for

insights. In a vision for next-generation monitoring infrastructure, Sanchez et al. [17] presents that visualization helps to monitor computing nodes to support a variety of tasks with parallel log and data collection, assisting multiple sources for data correlation and management, conducting analysis over time. Grafana [5] is an open-source analytics and monitoring solution commonly used for HPC systems. Built upon Grafana, Schwaller et al. [18] incorporate analysis-driven visualizations in an HPC system data pipeline, where the results of analyses and HPC-oriented summaries are presented in the Grafana front-end interface.

Regarding monitoring tools, Ganglia [10] is a scalable, distributed monitoring tool for high-performance computing systems and compute grids, such as the cluster of web servers. RRDtool [14] serves as a back-end database to Ganglia, which can be applied for off-system analysis and visualization. Nguyen and Dang [13] propose an interactive visual interface for real-time monitoring the health status of HPC systems, emphasizing on detection of abnormal events and their temporal and spatial correlations. Visualization can serve as the fourth and final stage, along with Collection, Storage and Analysis in PIKA, monitoring infrastructure proposed by Dietrich et al. [3], where recorded metadata and time-series data can be visualized at runtime or post-mortem, and data can then be stored for further long-term analysis.

Graph-based approaches in visualization have been used in monitoring HPC performance in previous literature. Haridasan and Pfitscher [6] propose an approach for cluster monitoring with graphical visualization for performance analysis of MPI applications, where the edition of vertices and edges represents tasks and communication. The tasks contain information associated such as rank, execution time, a path of the corresponding executable file, and the machine responsible for the execution, which is relatively similar to our approach. Event flow graphs are proved by Aguilar et al. [1] to use less storage reduction than regular event traces and allows the full ordered sequence of events performed by the application to be recovered. When the demands of monitoring require multiple monitors in a single virtual display, a graph-based approach is adapted [8] where graph viewers can be easily enhanced so that they can operate on virtual reality-enabled multi-monitor environments.

The principle of node positioning and the resulting flexibility in force-directed layout lead to its application in monitoring. Mansman et al. [9] present a visual analytics tool that visualizes network host behavior through positional changes in a two-dimensional space for a large dataset, using a force-directed graph layout algorithm. Upon the infrastructure of the cloud computing platform CLAVIRE, Zagarskikh et al. [23] presents a visualization provision tool for resource visualization, based on a modified force-directed layout to produce new force balance and geographical coordinates. Also incorporated with geo-location, this layout algorithm is used for community detection for situational analytics [12]. Cryptocurrency system is one of the emerging fields recently, resulting in the surveillance of the mining pools playing an essential role in such system. Xia et al. [22] incorporate the force-directed algorithm in providing interactions in a dynamic network for cross-view analysis and identity marking in mining pool surveillance.

In recent research, TRACER, a tool for visual exploration of network topology with 3D force-directed layout proposed by Tretyakov et al. [20], is applied to CERN computing infrastructure and can be deployed in near real-time network monitoring.

While graphs have previously been applied to monitoring tasks, the force-directed layout has limited use and has been unexplored in HPC monitoring. The novelty of our approach lies in its objective to make use of the flexibility of the force-directed algorithm in demonstrating the dynamic characteristics in the temporal and structural association among HPC system components. In this paper, we develop 2D and 3D visualization to better depict the internal structure and changes in topology over time. At the same time, user interaction facilitates in-depth analysis and supports details-on-demand through visual exploration.

### 3 Design and implementation

#### 3.1 Terms and definitions

In this section, we define terms that are related to the structural components inside the HPC system [2]. These components and their associations, reflecting the data structure, are presented as visual elements in the visualization of *JobNet*.

**Cluster:** The top-level organizational unit of an HPC cluster, comprising a set of nodes, a queue, and jobs.

**Node:** A single, named host (compute server) in the cluster.

**Task:** The execution of a user application.

**Job:** A resource request that is submitted for execution that contains one or more tasks. One job can run on one or more nodes.

**Queue:** A list of jobs that have been submitted to the service to be run by the cluster. A set of scheduling policies determines the order in which to run jobs from the queue.

**User:** An entity that submits a job for execution. A user is associated with a set of a running job at a particular time point.

#### 3.2 Design rationale

**Visualizing structural association** Monitoring the HPC system can be formulated as observational tasks and ultimately decision making on the temporal changes on the HPC system components. The dynamic nature of such structures as a network of users, jobs, compute nodes, and their associations can make the monitoring task more complex.

The basic building blocks, represented by the leaves in the graph, are the compute servers, also known as computing nodes. Each of these nodes is associated with the job that executes on it, where one job can have multiple nodes running simultaneously, and one compute server can run one or more jobs simultaneously. In a similar idea, each job is associated with the user submitting it; while one user can submit multiple jobs, each job only belongs to one user.

At a particular time point, a job can release the node after finishing, start or continue using a node.

Our design decisions are based on the visual information mantra by Shneiderman [19]: Overview first, zoom and filter, then details-on-demand. *JobNet* applies force-directed layout to demonstrate the aforementioned structural association between compute servers, jobs, and users. This design concept is applied for both 2D and 3D versions. The dynamic characteristics of this structure, or the network topology changes over time, are described in the next section.

To assist the task of performance monitoring, *JobNet* users can investigate every indicator within the health metrics, including CPU temperatures (of main processing unit CPU1 and supporting unit CPU2), Inlet temperature, Fan speed (per each of four fans for dissipating heated air away from the components), and Power consumption. The color scale is consistent across all health metrics, ranging from the corresponding minimum to maximum value.

**Time representation** There are various ways to incorporate time into the visualization. One common approach is to have the timeline as an axis and represent dependent visual elements along the timeline. However, with a large number of visual elements of the graph at a time point, visualizing all elements along the timeline would make the view cluttered and difficult to understand. This issue is addressed in 2D and 3D visualization as follows.

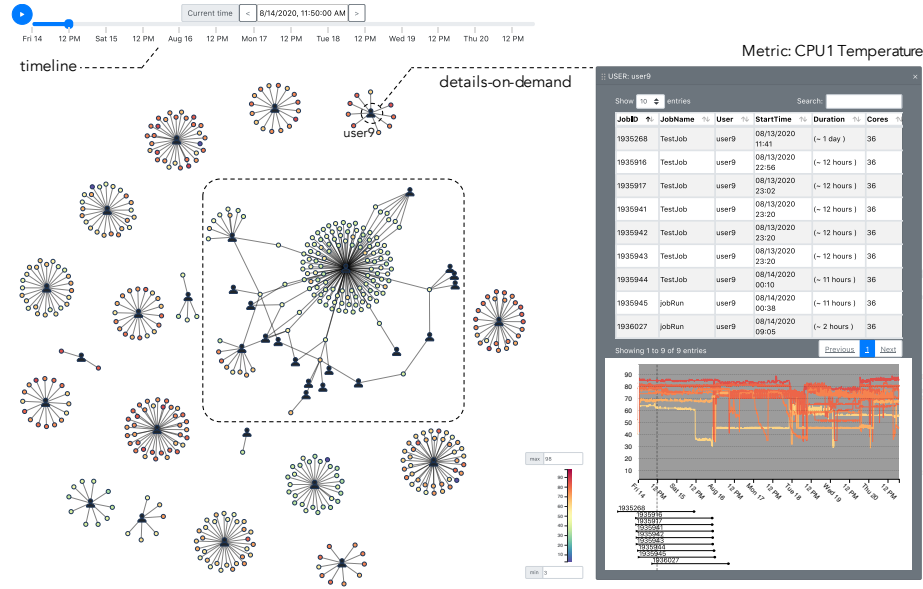
*Incorporating animation* In the 2D visualization, the timeline is combined with the time slider and a play/pause button. The changes in graph topology over time are made visible by the transition in animation. As default, the clock on the time slider runs automatically, and the corresponding topology will be updated accordingly on the graph. The view is coupled with a narration of the changes in nodes usage. Details-on-demand is provided upon mouse-click to support in-depth analysis into computing usage for the entire time.

*Visualizing changes exclusively* For the 3D visualization, we utilize the added dimension for time representation. Taking into account the issue of view cluttering, *JobNet* only visualizes the changes of node releasing or node starting to be used in the whole structure. This 3D representation allows 360-degree rotation so that the user can examine the entire graph from the beginning until the end of the observation period from a different angle for the complete view. For this 3D interface, we use canvas [11] to present the visual elements. Canvas is suitable for fast rendering when there are a large number of visual elements involved, as all elements drawn on a canvas element do not manifest in the DOM and save a substantial amount of work for the browser [21].

## 4 Case study

### 4.1 JobNet2D

The 2D visualization of *JobNet* is demonstrated in Figure 1. First, the user can have an overview of the current snapshot at the particular time point determined

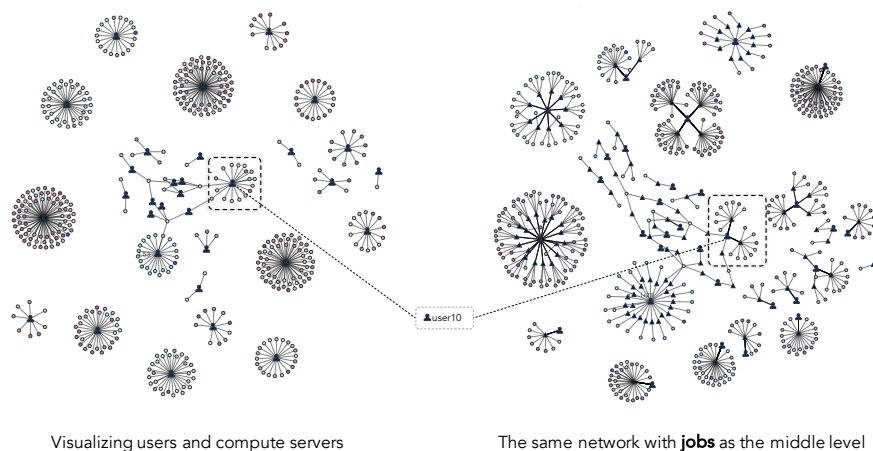


**Fig. 1.** 2D interface of JobNet. Details-on-demand is available upon mouse click.

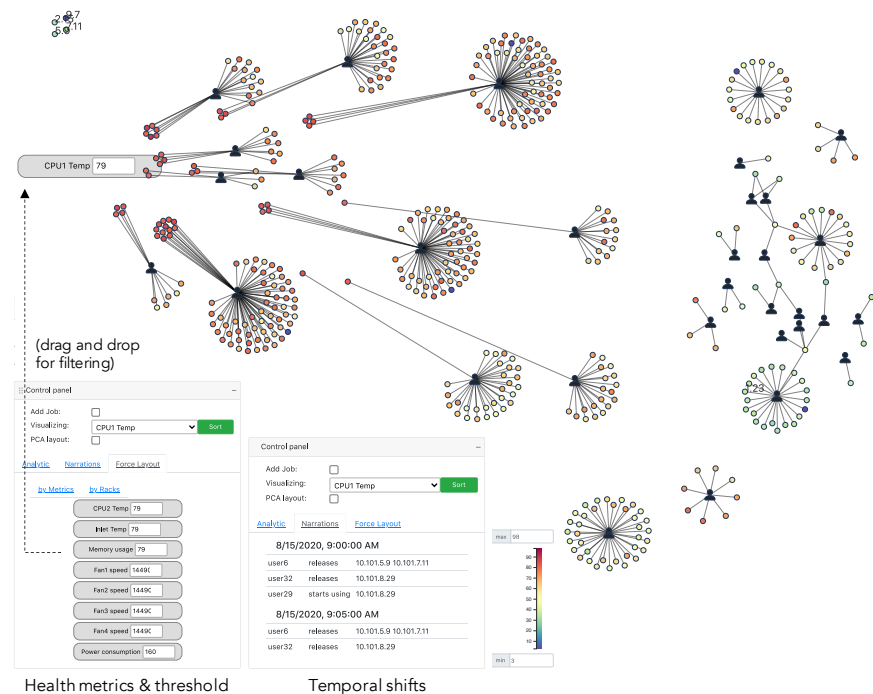
at the time slider incorporated with the timeline at the top. From the overview, we can see that the force-directed layout algorithm clearly presents the users with their corresponding compute nodes, including the cases where one compute is used by multiple users, which is emphasized in the dashed-line box in the middle of the graph in Figure 1. The current metric on displaying is CPU1 temperature.

The interface provides details-on-demand upon mouse clicking on the user. As the selection of *user9* is specified, the details of running jobs and their executing computes are shown on the pop-up panel in Figure 1. The details show that the compute nodes for *user9* have high indicators for CPU1 *JobNet* users can also zoom in the network for a close-up view at the graph vertices. On mouse hovering, the selection of the user or node will be displayed; other visual elements will be filtered. Upon drag and drop, the *JobNet* user can filter the nodes based on the corresponding thresholds set on health metrics, as depicted in Figure 3. The nodes whose value is greater than the threshold will be pulled closer to the metric box; otherwise, the nodes will be pushed away from the box.

For simplicity, the default presentation shows users and their running compute servers. *JobNet* user can also enable the visibility of jobs to show the order of association of the user-job-compute server. Figure 2 shows the difference between visualizing users and compute servers alone, with the same network, but with the addition of jobs as the middle level, *user10* and its set of the same compute servers are highlighted in two cases. Without the visibility of jobs, we cannot see that there are actually two jobs running on that group of nodes. This detail comes with a trade-off of network complexity and legibility.



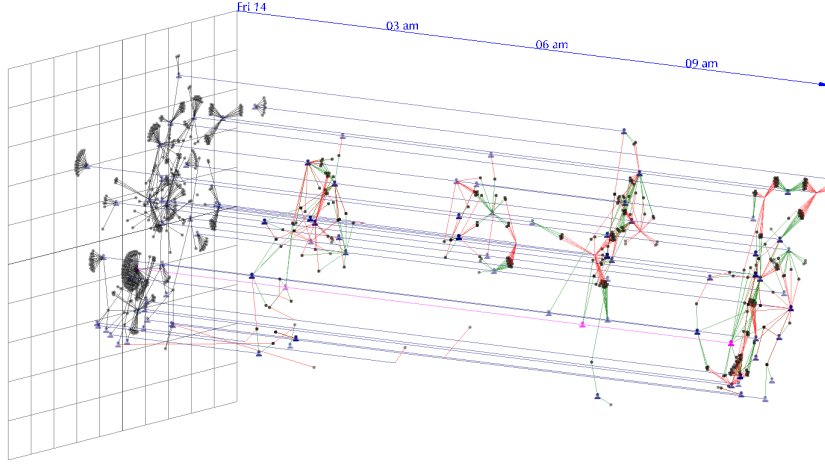
**Fig. 2.** The default option for visualizing users and compute servers (left), and the detailed option having jobs added as the middle level for the order of association of user-job-compute server.



**Fig. 3.** Interactions: Drag and drop health metrics boxes with their threshold into the graph area for intuitive filtering: The nodes whose value greater than the threshold will be pulled closer to the metric box, otherwise the nodes will be pushed away from the box.



## 4.2 JobNet3D

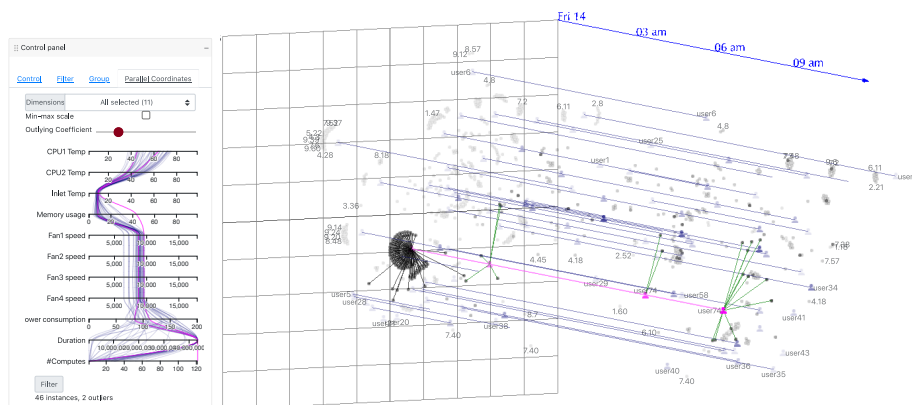


**Fig. 4.** 3D interface of JobNet.

Figure 4 demonstrate the 3D interface of *JobNet*. Instead of repeating the complete network topology in every time step, *JobNet* presents the visual elements carrying changes in status: a red line represents a node being released (job finished), whereas a green line corresponds to a node started to be put in execution. The spatial locations of the nodes and users remain consistent along the timeline. Compared to the 2D version, the 3D interface shows all changes that occur in the entire observational interval at once, instead of relying on *JobNet* user’s eyes activity and mental memory upon transition. In other words, the view provided by the 3D visualization indicates global evolution without cluttering the view, also helps users spot changes without spending too much mental energy on remembering the preceding state.

The visualization allows adjusting the time resolution (time compressing) for showing data for single time steps (every 5 minutes) or aggregation at 30 minutes, or 1, 2, or 3 hours for simplicity and demonstration purpose. The time resolution for the visualization in Figure 4 is 3 hours. The smaller window for aggregation results in more discrete data points along the timeline, leading to more precise positions but also a more cluttered view.

**Outlier detection** *JobNet* in 3D provides users with interactive filtering through specifying a range on the parallel coordinates in the control panel. Figure 5 shows *JobNet* visualization with an outlier highlighted upon mouse hovering. The outlier implementation for 3D representation is based on the work of Pham and Dang [16]. The said outlier, *user74* only has green links connected to it,



**Fig. 5.** Filtering options with *JobNet*. The filtering panel shows the parallel coordinates where outlier users are indicated. *JobNet* 3D interface is highlighted with mouse hovering interaction over an outlier.

meaning that the user executes on more and more compute nodes but never releases one.

**Applying on a Different Domain** This case study applies on PopCha! [15], an Android application that allows its users to follow their movies, discover movies with the recommendation and sharing feature. The dataset that we use in this work is a subset of the movie network produced from data in the PopCha! application. Movies related to each other as they get picked by PopCha! users. The dataset contains records for 51 years. Figure 6 demonstrates the use of *JobNet* on the PopCha! dataset. In terms of visual encodings, nodes in the graph are actors or actresses (1,113 elements), a link between two nodes shows that the two people are co-stars in a movie (6,200 links), while the color of a link indicates the genre of the movie they appear together.

Figure 7 depicts the pattern upon zooming for a close-up view. Within the captured time period, there are one-time patterns in two movies, *Apocalypse Now* and *Blade Runner* as circled in black and blue. There is also a repeated pattern, including actor Harrison Ford in all occurrences, which signifies the three sequels of the movie Star Wars.

## 5 Conclusion

In the scope of this paper, we discussed a novel visualization tool to monitor the temporal and structural association of HPC system components. The tool applies a force-directed layout algorithm on the 2D and 3D visualizations to represent the system users, jobs submitted, and compute servers for execution. Interactions on zooming, filtering, and detail-on-demand offer means for visual

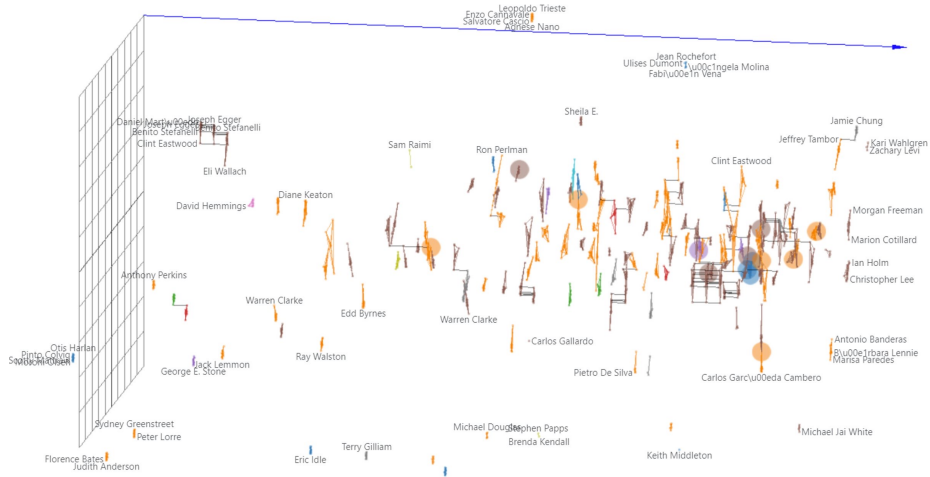


Fig. 6. 3D visualization of *JobNet* applied on the PopCha! dataset.

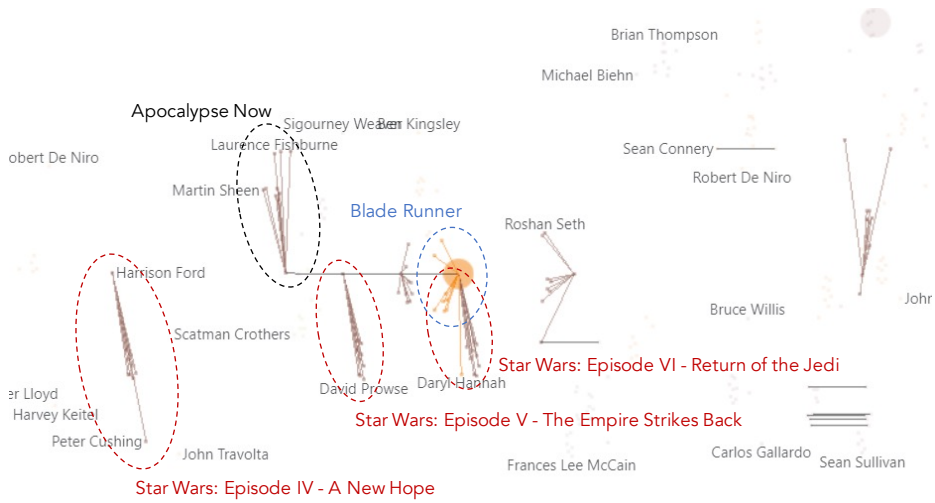


Fig. 7. Close-up view on the 3D visualization of *JobNet*. The repeated pattern is highlighted in red: A similar cast (inferred from the similar structures of the subnetworks) appear three times, indicating the three sequels of the movie Star Wars.

exploratory and support analysis. The usefulness of the tool is demonstrated by the case study on a real-world HPC job accounting dataset. Besides the HPC dataset, the 3D version of the tool shows its applicability to other domains and datasets having time constraints and mutual references among elements.

## Bibliography

- [1] Aguilar, X., Furlinger, K., Laure, E.: Visual mpi performance analysis using event flow graphs. *Procedia Computer Science* **51**, 1353–1362 (2015)
- [2] Dan Lepow, J.L.: Overview of hpc job manager (2016), <https://docs.microsoft.com/en-us/powershell/high-performance-computing/overview-of-hpc-job-manager?view=hpc19-ps>
- [3] Dietrich, R., Winkler, F., Knüpfer, A., Nagel, W.: Pika: Center-wide and job-aware cluster monitoring. In: 2020 IEEE International Conference on Cluster Computing (CLUSTER). pp. 424–432. IEEE (2020)
- [4] Fruchterman, T.M., Reingold, E.M.: Graph drawing by force-directed placement. *Software: Practice and experience* **21**(11), 1129–1164 (1991)
- [5] Grafana: The open platform for beautiful analytics and monitoring (2019), <https://grafana.com/>
- [6] Haridasan, M., Pfitscher, G.H.: Pm/sup 2/p: a tool for performance monitoring of message passing applications in cots pc clusters. In: Proceedings. 15th Symposium on Computer Architecture and High Performance Computing. pp. 218–225. IEEE (2003)
- [7] Holten, D., Van Wijk, J.J.: Force-directed edge bundling for graph visualization. In: *Computer graphics forum*. vol. 28, pp. 983–990. Wiley Online Library (2009)
- [8] Jingai, R., Kido, Y., Date, S., Shimojo, S.: Research note: A high resolution graph viewer for multi-monitor visualization environment. *The Review of Socionetwork Strategies* **9**(1), 19–27 (2015)
- [9] Mansman, F., Meier, L., Keim, D.A.: Visualization of host behavior for network security. In: *VizSEC 2007*, pp. 187–202. Springer (2008)
- [10] Massie, M.L., Chun, B.N., Culler, D.E.: The ganglia distributed monitoring system: design, implementation, and experience. *Parallel Computing* **30**(7), 817–840 (2004)
- [11] Network, M.D.: Canvas tutorial (2021), [https://developer.mozilla.org/en-US/docs/Web/API/Canvas\\_API/Tutorial](https://developer.mozilla.org/en-US/docs/Web/API/Canvas_API/Tutorial)
- [12] Nguyen, H.N., Dang, T.: EQSA: Earthquake Situational Analytics from Social Media. In: 2019 IEEE Conference on Visual Analytics Science and Technology (VAST). pp. 142–143 (2019). <https://doi.org/10.1109/VAST47406.2019.8986947>
- [13] 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). PEARC '19, Association for Computing Machinery, New York, NY, USA (2019). <https://doi.org/10.1145/3332186.3337959>, <https://doi.org/10.1145/3332186.3337959>
- [14] Oetiker, T.: Rrdtool (2014), <http://rrdtool.org>
- [15] PDI, T.: Popcha! movies, tv & theaters (2013), <https://popcha-movies-tv-amp-theaters.soft112.com/>

- [16] Pham, V., Dang, T., Wilkie, A., Banterle, F.: Scagnosticsjs: Extended scatterplot visual features for the web. In: Eurographics (Short Papers). pp. 77–80 (2020)
- [17] Sanchez, S., Bonnie, A., Van Heule, G., Robinson, C., DeConinck, A., Kelly, K., Snead, Q., Brandt, J.: Design and implementation of a scalable hpc monitoring system. In: 2016 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW). pp. 1721–1725. IEEE (2016)
- [18] Schwaller, B., Tucker, N., Tucker, T., Allan, B., Brandt, J.: Hpc system data pipeline to enable meaningful insights through analysis-driven visualizations. In: 2020 IEEE International Conference on Cluster Computing (CLUSTER). pp. 433–441. IEEE (2020)
- [19] Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings 1996 IEEE Symposium on Visual Languages. pp. 336–343 (1996). <https://doi.org/10.1109/VL.1996.545307>
- [20] Tretyakov, E., Artamonov, A., Grigorieva, M., Klimentov, A., McKee, S., Vukotic, I.: Tracer (trace route explorer): A tool to explore osg/wlwg network route topologies. *International Journal of Modern Physics A* **36**(5), 2130005–10 (2021)
- [21] Verspohl, L.: D3 and canvas in 3 steps (2017), <https://www.freecodecamp.org/news/d3-and-canvas-in-3-steps-8505c8b27444/>
- [22] Xia, J.z., Zhang, Y.h., Ye, H., Wang, Y., Jiang, G., Zhao, Y., Xie, C., Kui, X.y., Liao, S.h., Wang, W.p.: Supoolvisor: a visual analytics system for mining pool surveillance. *Frontiers of Information Technology & Electronic Engineering* **21**(4), 507–523 (2020)
- [23] Zagarskikh, A., Karsakov, A., Mukhina, K., Nasonov, D., Bezgodov, A.: An efficient approach of infrastructure processing visualization within cloud computing platform. *Procedia Computer Science* **66**, 705–710 (2015)