Visualizing Multidimensional Health Status of Data Centers

Tommy Dang(✉)

Texas Tech University, Lubbock, TX 79409, USA
tommy.dang@ttu.edu
http://www.myweb.ttu.edu/tnhondan

Abstract. Monitoring data centers is challenging due to their size, complexity, and dynamic nature. This project proposes a visual approach for situational awareness and health monitoring of high-performance computing systems. The visualization requirements are expanded on the following dimensions: (1) High performance computing spatial layout, (2) Temporal domain (historical vs. real-time tracking), and (3) System health services such as temperature, CPU load, memory usage, fan speed, and power consumption. To show the effectiveness of our design, we demonstrate the developed prototype on a medium-scale data center of 10 racks and 467 hosts. The work was developed using feedback from both industrial and acadamic domain experts.

Keywords: Scatterplots · Visual features · High-dimensional data · Data center visualization · High-performance computing systems · Redfish RESTful API · Nagios Core · Baseboard management controller

1 Introduction

Data centers are increasingly complex and hence monitoring such systems is a daunting task for system administrators. In 2013, the Distributed Management Task Force (DMTF) released *Redfish* [17], an open industry standard specification for server configuration that aims to supersede IPMI over the network (IPMI over LAN). The web servers are expected to provide end users with simple, secure management of scalable platform hardware by enhancing security and reliability to Baseboard Management Controller (BMC). Redfish uses industry standard RESTful API architecture over Hyper Text Transfer Protocol Secure (HTTPS) using JavaScript Object Notation (JSON) format based on Open Data Protocol (OData) as depicted in Fig. 1. Redfish-enabled technologies enable IT systems to operate in harmony. Most importantly, a Redfish-enabled system delivers, and single API for everything [13]. One of the critical advantages of REST APIs is that it introduces a great deal of flexibility, allowing any software clients with RESTful capabilities to interact with *Redfish* firmware.

Building on top of Redfish RESTful interface, this paper introduces a visual analytic prototype for monitoring high-performance computing (HPC) system

© Springer Nature Switzerland AG 2019

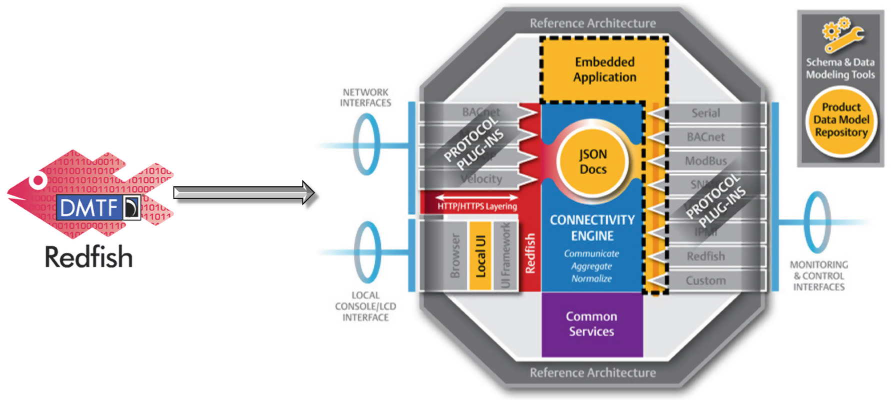


Fig. 1. Redfish reference architecture [13].

events. The goals of this prototype are: (1) to monitor the multidimensional health status of multiple hosts and racks in real-time, (2) to support system administrators in detecting unusual correlation of these services in a complex and highly dynamic system, and (3) to help in performing system troubleshooting and debugging. We demonstrate our prototype at a data center at a university.

The rest of this paper is organized as follows: We first summarize existing techniques for visualizing HPC in the next Section. Then we provide an overview of visualization tasks and describe the design and supported interactive features of our prototype in detail. In Sect. 4, we present the results of an informal study with HPC experts from a university and an industrial company. We argue that using visual features of pairwise projection we can capture abnormal correlations between various health dimensions in the data center. More importantly, these visual features can be computed automatically and notify the system administrator significant events. Finally, we conclude our paper with future plans for visual monitoring and predicting through the integration with Nagios and Redfish API.

2 Existing Approaches

Although not fully explored, there are many existing tools that support monitoring high-performance computing systems. For example, LLView [14] is a client-server based application which allows monitoring the utilization of clusters controlled by batch systems like IBM LoadLeveler, PBSpro, Torque, or IBM Blue Gene system database. However, multidimensional analysis proposed in this work is not in focus of LLview.

Ganglia [16] is an open source PHP-based web front-end interface that allows users to gather information via real-time dynamic web pages. This information, including CPU usage, memory usage, disk usage, network statistics, the number of running processes, is plotted on similar graphs. It leverages the widely used XML technologies for data representation. The advantage of real-time responsive

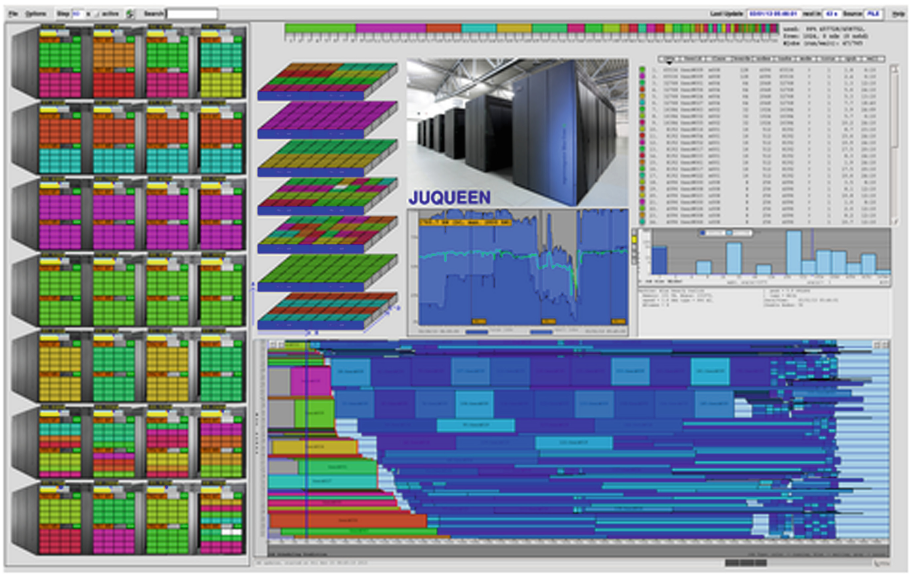


Fig. 2. Main window of LLview [14]: graphical monitoring of batch system controlled cluster.

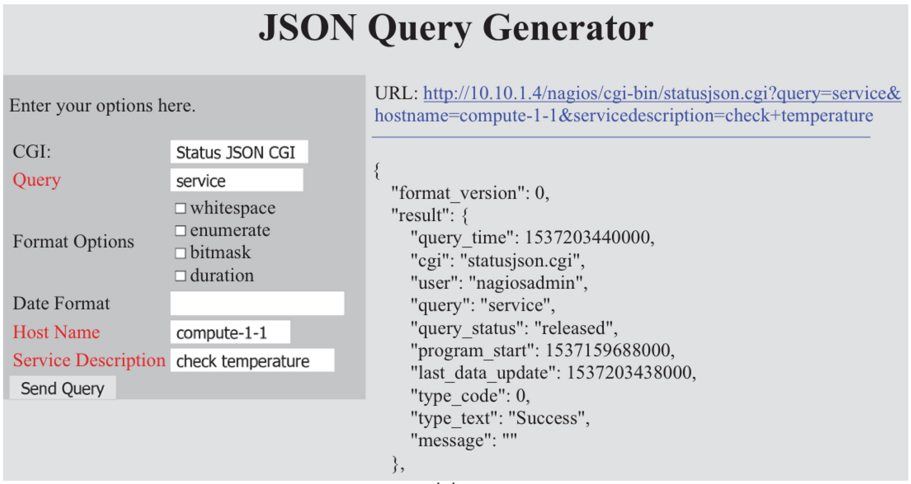
on the web front-end, however, leads to high latency due to the size of the XML tree. Thus, Ganglia is not practical on the less powerful machines when the amount of data is large (Fig. 2).

Nagios Core [6] is another open-source monitoring system that is capable of handling a variety of servers and operating systems with the industry standard. It provides a primary web interface for the core monitoring engine as depicted in Fig. 3(b). To trace a problem, however, an administrator cannot capture a holistic monitoring view by using the Nagios web interface. Even though basic filtering operations are provided, system status overview, which is useful to correlate the isolated temporal/spatial issues, can be lost [5,11]. Figure 3(a) shows an example of the JSON query on CPU temperatures (on the left) and the corresponding results (on the right) returned from the server.

These existing tools inspect system status independently. This paper takes into account the correlations between various dimensions such as CPU temperature, memory usage, fan speed, power consumption, and I/O bandwidth. Analyzing the relationships of multidimensional data is essential to understand system behaviors as a whole [2].

3 Visualization Components

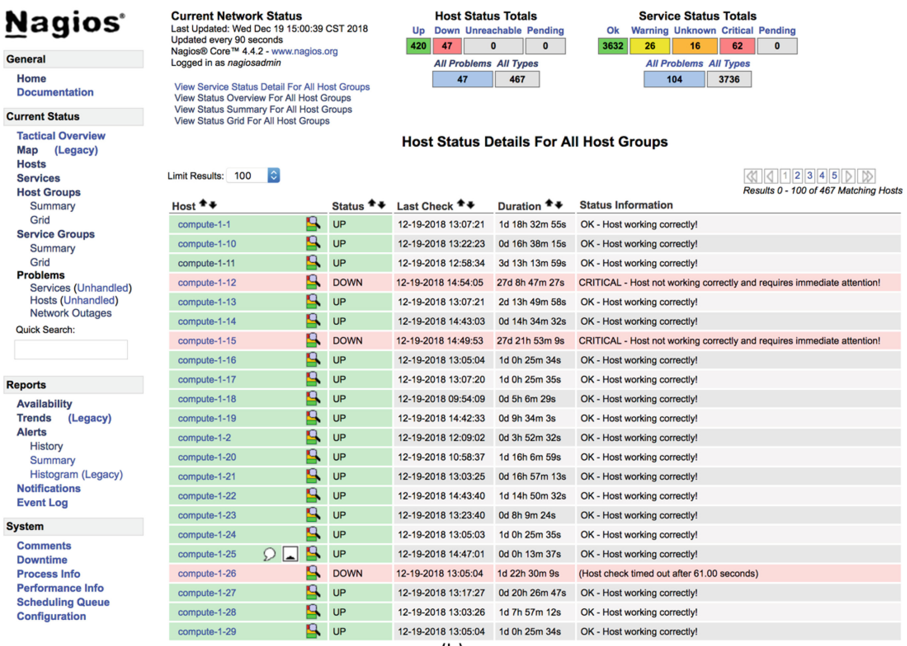
Through in-depth discussions with domain experts, we have identified the set of design goals: (1) Provides spatial and temporal overview across hosts and racks, (2) Allows system administrators to quickly narrow down to the event of interest



CGI:	Status JSON CGI
Query	service
Format Options	☐ whitespace ☐ enumerate ☐ bitmask ☐ duration
Date Format	
Host Name	compute-1-1
Service Description	check temperature

URL: http://10.10.1.4/nagios/cgi-bin/statusjson.cgi?query=service&hostname=compute-1-1&servicedescription=check+temperature

```
{
  "format_version": 0,
  "result": {
    "query_time": 1537203440000,
    "cgi": "statusjson.cgi",
    "user": "nagiosadmin",
    "query": "service",
    "query_status": "released",
    "program_start": 1537159688000,
    "last_data_update": 1537203438000,
    "type_code": 0,
    "type_text": "Success",
    "message": ""
  },
```



Current Network Status
Last Updated: Wed Dec 19 15:00:39 CST 2018
Updated every 90 seconds
Nagios® Core™ 4.4.2 - www.nagios.org
Logged in as *nagiosadmin*

View Service Status Detail For All Host Groups
View Status Overview For All Host Groups
View Status Summary For All Host Groups
View Status Grid For All Host Groups

Host Status Totals

Up	Down	Unreachable	Pending
420	47	0	0

All Problems	*All Types*
47	467

Service Status Totals

Ok	Warning	Unknown	Critical	Pending
3632	26	16	62	0

All Problems	*All Types*
104	3736

Host Status Details For All Host Groups

Limit Results: 100

1 2 3 4 5
Results 0 - 100 of 467 Matching Hosts

Host	Status	Last Check	Duration	Status Information
compute-1-1	UP	12-19-2018 13:07:21	1d 18h 32m 55s	OK - Host working correctly!
compute-1-10	UP	12-19-2018 13:22:23	0d 16h 38m 15s	OK - Host working correctly!
compute-1-11	UP	12-19-2018 12:58:34	3d 13h 13m 59s	OK - Host working correctly!
compute-1-12	DOWN	12-19-2018 14:54:05	27d 8h 47m 27s	CRITICAL - Host not working correctly and requires immediate attention!
compute-1-13	UP	12-19-2018 13:07:21	2d 13h 49m 58s	OK - Host working correctly!
compute-1-14	UP	12-19-2018 14:43:03	0d 14h 34m 32s	OK - Host working correctly!
compute-1-15	DOWN	12-19-2018 14:49:53	27d 21h 53m 9s	CRITICAL - Host not working correctly and requires immediate attention!
compute-1-16	UP	12-19-2018 13:05:04	1d 0h 25m 34s	OK - Host working correctly!
compute-1-17	UP	12-19-2018 13:07:20	1d 0h 25m 35s	OK - Host working correctly!
compute-1-18	UP	12-19-2018 09:54:09	0d 5h 6m 29s	OK - Host working correctly!
compute-1-19	UP	12-19-2018 14:42:33	0d 9h 34m 3s	OK - Host working correctly!
compute-1-2	UP	12-19-2018 12:09:02	0d 3h 52m 32s	OK - Host working correctly!
compute-1-20	UP	12-19-2018 10:58:37	1d 16h 6m 59s	OK - Host working correctly!
compute-1-21	UP	12-19-2018 13:03:25	0d 16h 57m 13s	OK - Host working correctly!
compute-1-22	UP	12-19-2018 14:43:40	1d 14h 50m 32s	OK - Host working correctly!
compute-1-23	UP	12-19-2018 13:23:40	0d 8h 9m 24s	OK - Host working correctly!
compute-1-24	UP	12-19-2018 13:05:03	1d 0h 25m 35s	OK - Host working correctly!
compute-1-25	UP	12-19-2018 14:47:01	0d 0h 13m 37s	OK - Host working correctly!
compute-1-26	DOWN	12-19-2018 13:05:04	1d 22h 30m 9s	(Host check timed out after 61.00 seconds)
compute-1-27	UP	12-19-2018 13:17:27	0d 20h 26m 47s	OK - Host working correctly!
compute-1-28	UP	12-19-2018 13:03:26	1d 7h 57m 12s	OK - Host working correctly!
compute-1-29	UP	12-19-2018 13:05:04	1d 0h 25m 34s	OK - Host working correctly!

Fig. 3. Nagios Core [6] interfaces: (a) System summary in a simple listing format and (b) Json query on CPU temperatures and the corresponding results.

for system debugging, and (3) Inspects the correlation of system health services in a single view. To meet these goals, this paper proposes several visualization tasks:

- **Overview Display (T1).** Display an overview of real-time system status on the corresponding spatial layout [15].
- **Details-On-Demand (T2).** Users can inspect multidimensional historical data of a host in the system via a simple click [18].
- **Filtering (T3).** Highlight critical events on a host [3] and the associated time stamps [12].
- **Multidimensional analysis (T4).** Explore the correlations between dimensions [4,10] such as fan speeds, memory usage, and power consumption.

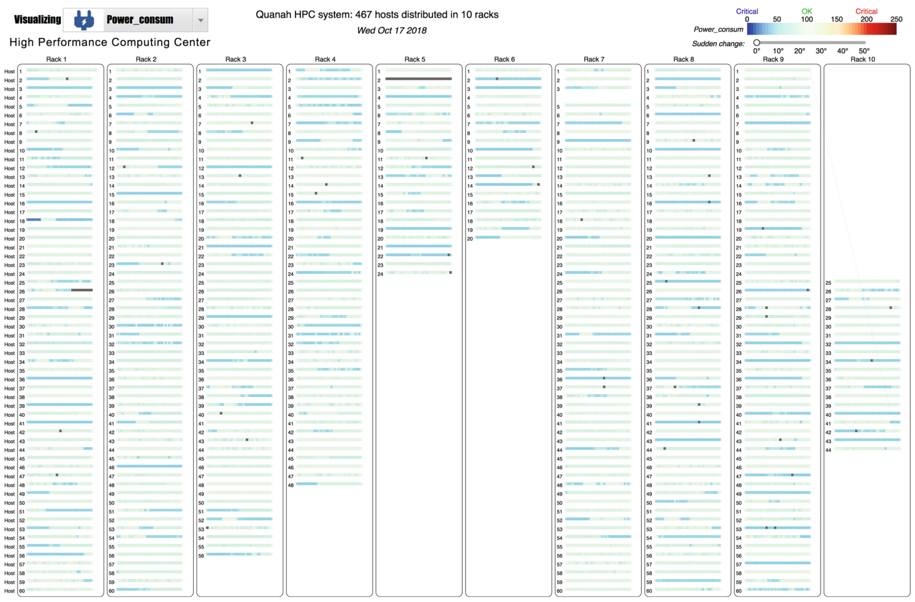


Fig. 4. Visualizing power consumption on Wednesday, October 17, 2018, of the 467-node Quanah cluster at Texas Tech University.

We leverage the Nagios Core engine for data retrieval through a RESTful API web interface. In other words, we iteratively request health status of every host int he system. For each host, we obtain a set of updated status as shown on the right panel of Fig. 3(a). In the next section, we present our approach for displaying status updates of hundreds of hosts within a single view. In the next section, we first introduce the HPC system at Texas Tech University and then visualization components and the supported interactions.

3.1 HPC System Spatial Layout

Figure 4 shows a snapshot power consumption of the 467-node Quanah cluster at Texas Tech University at noon on Sunday, May 13, 2018 (the visualization task

T1). Within a rack, hosts are listed top down. Rows are power consumption time series (the temporal distance between two consecutive cells is 3 min, the updating period set up on the cluster. Rectangles on each row are the computed power consumption, which is colored: red is high power consumption nodes while blue is low power consumption nodes.

3.2 Multidimensional Analysis of Health Status

Through our discussions with domain experts, it is essential to have a holistic view of the entire data center on multiple dimensions for system monitoring and especially diagnosis. Therefore, our system supports a multi-axis visualization of hosts on demand (visualization task **T2**). In particular, when users mouse over a host, a pop-up window is displayed to unveil the details of host information as depicted in Fig. 5. Historical temperatures of each CPU in the selected host are presented in line graphs: The vertical axis represents the CPU temperature (from 0 to 100° of Fahrenheit) while the horizontal axis represents the time.

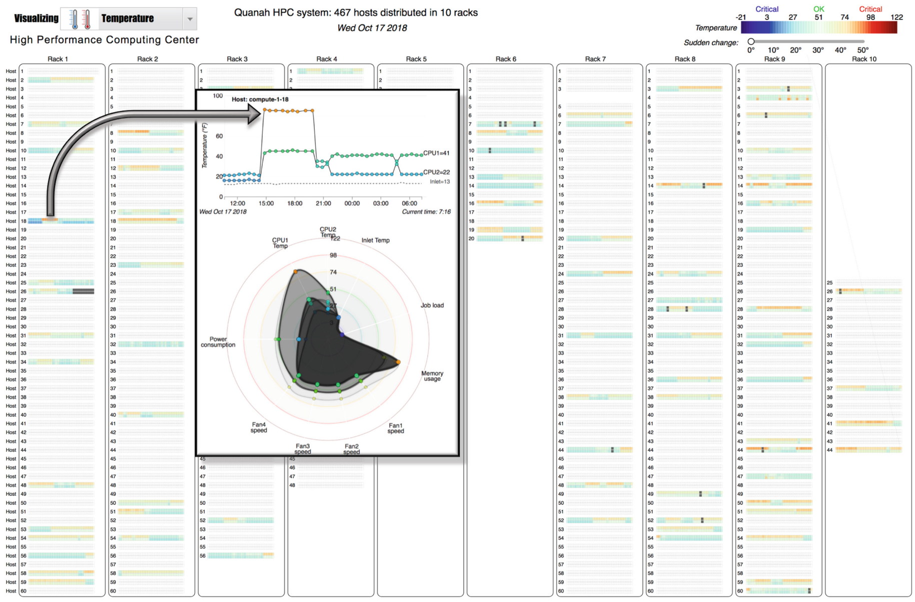


Fig. 5. Visualizing CPU temperature on Wednesday, October 17, 2018 of the 467-node Quanah cluster. Hosts experiencing a sudden temperature increase of 25° in 3 min.

The spider or cobweb chart [7] in the lower panel of the pop-up window shows health dimensions of the selected host (visualization task **T4**) which are grouped by category: CPU temperature (degrees Fahrenheit), job load (number of jobs assigned on the the given host), memory usage (from 0% to 100%), fan speeds, and power consumption (in watts). The purpose of this chart is to enable users to detect similar multidimensional patterns and how they change over time.

We decide to use the spider chart to display multidimensional health status of data center since the mental images of spider charts allow users to capture and compare historical data between different hosts quickly. Figure 6 shows four examples of spider charts from four random hosts in the Quanah cluster at Texas Tech University. In particular, each closed curve on the spider charts represents a multidimensional observation. The observed period is from 11 am to 7 pm on Wednesday, October 17, 2018.

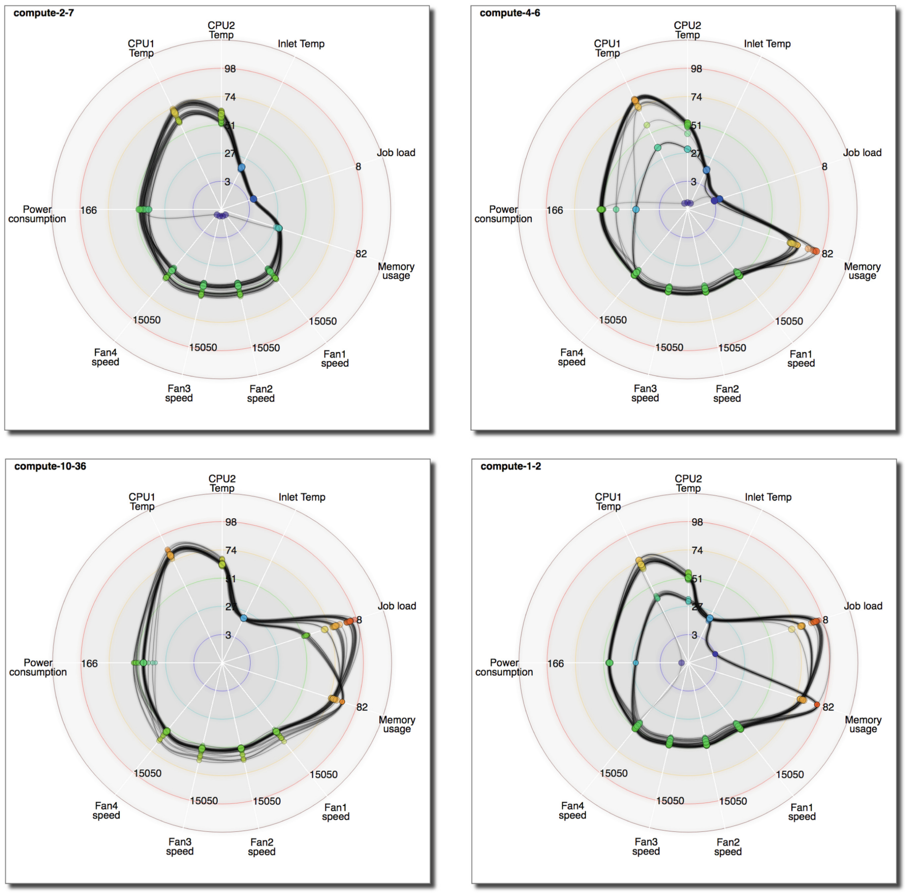


Fig. 6. Visualizing multidimensional heath status of four hosts in the data center.

Users can request to show multidimensional spider chart across hosts in the system at a time stamp. Figure 7 shows our multidimensional visualization of 467 hosts at Texas Tech University at 2 pm on Wednesday, October 17, 2018.

Brushing and linking between the summary view and HPC spatial layout can be done via *mouse over*. Our prototype also supports a range of interactive features, such as zooming and filtering (visualization task **T3**). In Fig. 5, we only display hosts which experienced a sudden CPU heat up of over 25° Fahrenheit within the two consecutive time stamps.

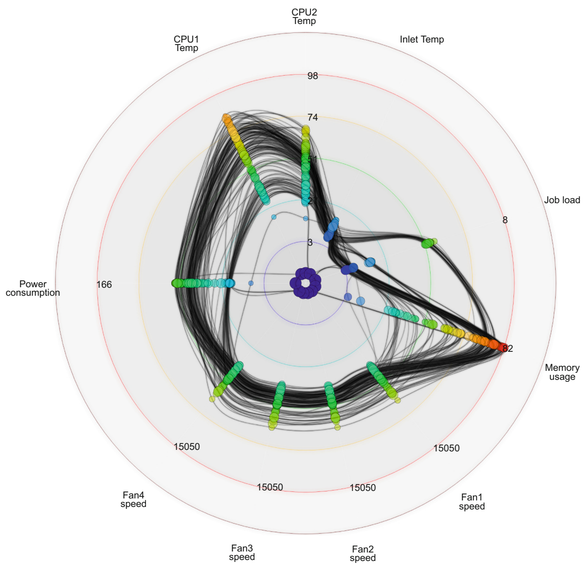


Fig. 7. Visualizing multidimensional heath status at 2 pm on Wednesday, October 17, 2018, of the 467-node Quanah cluster at Texas Tech University.

4 Discussion and Future Work

Our multidimensional analysis approach received positive feedback from both industrial and academic domain experts and encouraged to explore further. One potential direction is inspecting pairwise correlations within a 2D projection [19]. Figure 8 show an example of the 467-node Quanah cluster projected on fan speed (x axis) vs. CPU temperature (y axis). The plot reveals two clusters along the x axis and an outlier (the *compute 4–17*) at the lower left corner which has low fan speed and temperature.

Instead of having a human looking into every single scatterplot, visual features [20] can help to simplify this process by highlighting only unusual correlations. The following examples in Fig. 9 summarize possible 2D projections that can be captured using these visual features [9].

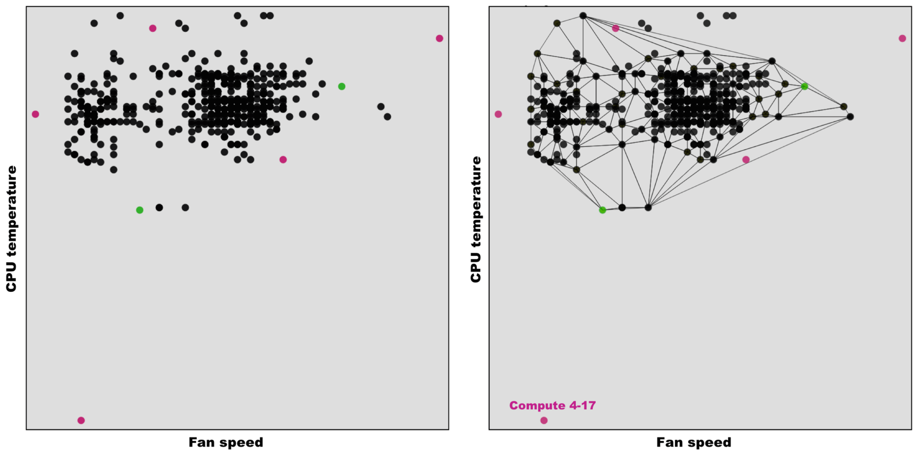


Fig. 8. Visualizing multidimensional heath status of all 467 hosts in the Quanah cluster at Texas Tech University: *Fan speed* vs. *CPU temperature*.

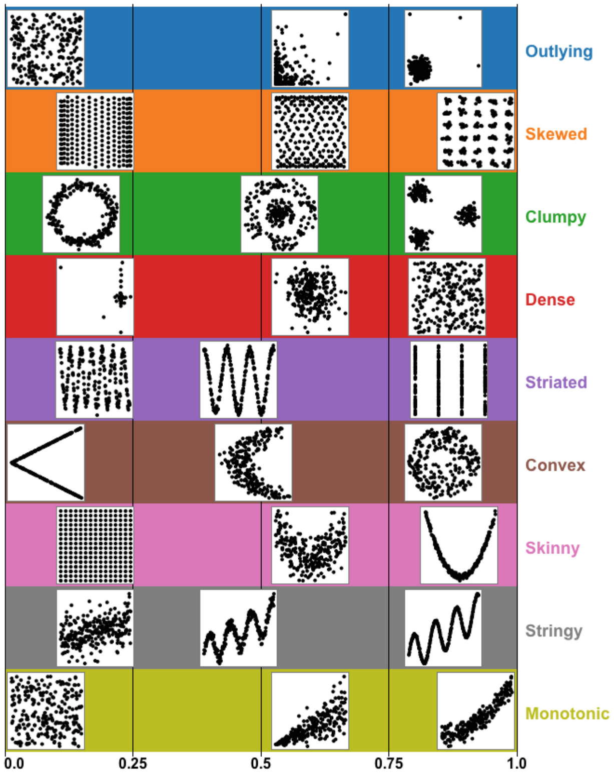


Fig. 9. Some example scatterplots and their measures [8]. In each row, the scatterplots with a low score on the associated measures are on the left while the scatterplots with a high score on the associated visual features are on the right.

5 Conclusion

This paper present a graphical tool for situational awareness and multidimensional health status of data centers using real-time data gathered through the industry-standard Redfish protocol and API (v.s. the aging IMPI protocol). The system has two components: visual feature extraction for pairwise projection and spider chart for visualizing high-dimensional health status. Our prototype supports a wide range of interactive features such as brushing and linking, zooming, and filtering.

In future work, we want to incorporate machine learning framework, such as TensorFlow [1] to predict the system health status. The model will be trained on historical data, make real-time predictions, and raise the alarm to the system administrator for timely actions. Google recently released tensorflow.js (https://js.tensorflow.org) which can be naturally adapted in our project. Virtual Reality and Augmented Reality (VR and AR) are also interesting future extensions. VR and AR interfaces enable real-time monitoring beyond traditional displays by experiencing interactive visualization techniques on mobile devices as well as within immersive environments.

References

1. Abadi, M., et al.: TensorFlow: large-scale machine learning on heterogeneous systems (2015). https://www.tensorflow.org/
2. Allcock, W., Felix, E., Lowe, M., Rheinheimer, R., Fullop, J.: Challenges of HPC monitoring. In: SC 2011: Proceedings of 2011 International Conference for High Performance Computing, Networking, Storage and Analysis, pp. 1–6, November 2011. https://doi.org/10.1145/2063348.2063378
3. Amar, R., Eagan, J., Stasko, J.: Low-level components of analytic activity in information visualization. In: Proceedings of the IEEE Symposium on Information Visualization, pp. 15–24 (2005)
4. Anand, A., Wilkinson, L., Dang, T.N.: Visual pattern discovery using random projections. In: 2012 IEEE Conference on Visual Analytics Science and Technology (VAST), pp. 43–52 (2012). https://doi.org/10.1109/VAST.2012.6400490
5. Andrienko, N., Andrienko, G., Gatalsky, P.: Exploratory spatio-temporal visualization: an analytical review. J. Vis. Lang. Comput. **14**(6), 503–541 (2003)
6. Barth, W.: Nagios: System and Network Monitoring. No Starch Press, San Francisco (2008)
7. Bremer, N.: A different look for the D3.js radar chart (2015). https://www.visualcinnamon.com/2015/10/different-look-d3-radar-chart. Accessed Oct 2018
8. Dang, T., Wilkinson, L.: Scagexplorer: exploring scatterplots by their scagnostics. In: Proceedings of the 2014 IEEE Pacific Visualization Symposium, PACIFICVIS 2014, pp. 73–80. IEEE Computer Society, Washington (2014). https://doi.org/10.1109/PacificVis.2014.42
9. Dang, T., Wilkinson, L.: Transforming scagnostics to reveal hidden features. IEEE Trans. Vis. Comput. Graph. **20**(12), 1624–1632 (2014). https://doi.org/10.1109/TVCG.2014.2346572

10. Dang, T.N., Anand, A., Wilkinson, L.: TimeSeer: scagnostics for high-dimensional time series. IEEE Trans. Vis. Comput. Graph. **19**(3), 470–483 (2013). https://doi.org/10.1109/TVCG.2012.128
11. Dang, T.N., Pendar, N., Forbes, A.G.: TimeArcs: visualizing fluctuations in dynamic networks. Comput. Graph. Forum (2016). https://doi.org/10.1111/cgf.12882
12. Dang, T.N., Wilkinson, L.: TimeExplorer: similarity search time series by their signatures. In: Bebis, G., et al. (eds.) ISVC 2013. LNCS, vol. 8033, pp. 280–289. Springer, Heidelberg (2013). https://doi.org/10.1007/978-3-642-41914-0_28
13. of the Industry, V.: How redfish specifications can improve simplicity, visibility and control of it (2016). https://datacenterfrontier.com/redfish-specifications/. Accessed Oct 2018
14. Karbach, C., Valder, J.: System monitoring: LLview. In: Computational Science and Mathematical Methods. 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), 26–27 Nov 2015, Jülich, Germany (2015). https://juser.fz-juelich.de/record/279901
15. Keim, D.A., Panse, C., Sips, M.: Information visualization: scope, techniques and opportunities for geovisualization. In: Dykes, J. (ed.) Exploring Geovisualization, pp. 1–17. Elsevier, Oxford (2004)
16. Massie, M.L., Chun, B.N., Culler, D.E.: The ganglia distributed monitoring system: design, implementation, and experience. Parallel Comput. **30**(7), 817–840 (2004)
17. Organization, S.D.: Distributed management task force (2013). https://www.dmtf.org/standards/redfish
18. Shneiderman, B.: The eyes have it: a task by data type taxonomy for information visualizations. In: Proceedings of the 1996 IEEE Symposium on Visual Languages, VL 1996, p. 336. IEEE Computer Society, Washington (1996). http://dl.acm.org/citation.cfm?id=832277.834354
19. Wilkinson, L., Anand, A., Grossman, R.: Graph-theoretic scagnostics. In: Proceedings of the IEEE Information Visualization 2005, pp. 157–164. IEEE Computer Society Press (2005)
20. Wilkinson, L., Anand, A.: Grossman: D3 data-driven documents. IEEE Trans. Vis. Comput. Graph. **17**(12), 2301–2309 (2011)