

3D visualized data management for network devices based on multilevel caches

Chenyang Jiang, Yun Chen, Dongsheng Cai^{*}, Qi Huang

^a School of energy science and engineering, university of electronic science and technology of China, Chengdu, China

Abstract

Three dimensional visualization technologies are dramatically improving convenience and efficiency in modern industrial managements. In this paper, a novel multilevel caches-based 3D visualization system for managing device data, which could be applied for displaying 3D scenes and processing device data simultaneously, is investigated. The system is fast-response, compared to conventional strategies. This work may provide helpful references and inspirations for applications about computer based visualization, complex data management and network communications.

Keywords: 3D visualization, network devices, multilevel cache, data management

1. Introduction

Three-dimensional management with remote monitoring and direct operation has become an important research direction in recent years. It has great potential in lots of domain, such as smart grid, intelligent logistics and intelligent transportation [1]-[3]. In practical applications, considering that the devices in systems are unavoidably in different life period, distributed locations and complex topologies, it is challenging to manage equipment clusters and their data efficiently [4]-[5]. Currently, due to the dramatic progress of the networks based computer information processing technology, there are novel choices to manage complex systems, e.g. cloud computing technology [6], machine to machine strategy [7], and virtualized data center [8]. Accordingly, a highly intuitive and convenient method to visualize the devices would play a key role in future information processing [9]. However, most existing visualization techniques tend to focus on how to render scenes clearly and quickly, lacking in-depth study on how to form an efficient data management in the 3D visualized system simultaneously. Hence, a structured data management technique with high efficiency is in need [10]-[12].

In this paper, by using the Browser/Server (B/S) and J2EE, the authors propose a multilevel caches-based 3D visualized data managing scheme for network device management. In this scheme, the server side processes and computes complicate information, while the client sides are only responsible for receiving data request from users and information presentation. Accordingly, the conventional programs' limitation at the client side could be effectively avoided [13]. It also provides the users with high efficiency and convenience with dynamic interface. Moreover, taking advantage of the hierarchical display and multilevel cache, it is also time-saving, especially for concurrent accessing and cross-platform applications. Experimental results verify that adopting our technique, the 3D display speed reaches to about 0.005 s per device for one user, and about 0.04 s per device for 20 users simultaneously. The work presented here may be a good reference for visualizations, remote monitoring and distributed equipment managements.

* Manuscript received June 18, 2014; revised July 22, 2014.

Corresponding author: Dongsheng Cai; *E-mail address*: cai.dsheng@gmail.com.

doi: 10.12720/sgce.4.1.59-64

2. Design and Implement for 3D Information Display

The strategy of the proposed 3D visualized data management is schematically shown in Fig. 1. Once a user submits a data query for a certain device via browser, the server would accept it and try to address the device via local routing. Then the device would return its corresponding information to the server. Subsequently, the server would send the data back via Internet. Finally, the device would be three-dimensionally visualized at the browser side.



Fig. 1. The schematic diagram of the 3D visualized data management.

2.1. Integration for the device data

The data packet of a random device in the system is composed of 3 components, the image data, the real-time data and the static data, as shown in Fig. 2. Fig. 2 (a) presents the image of a device. We render a device by drawing each face of it with 2D image. This process makes the users operate the system conveniently with compressed data. Fig. 2 (b) shows the real-time data, such as temperature, IP, device status, etc, obtained from sensors or remote databases. Fig. 2 (c) shows the static data of the device, such as its brand, version and equipment number. Finally, the three parts are packaged together as a frame in Fig. 2 (d). In practical applications, a series of frames similar to Fig. 2 (d) are linked together to form a data center.

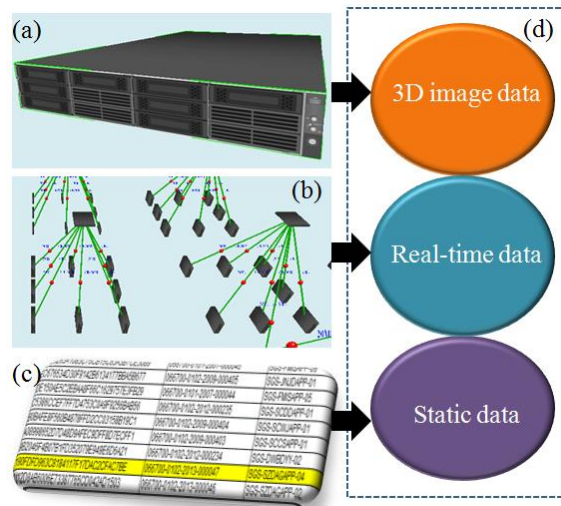


Fig. 2. Device data and integration: (a) the 3D information of the device, (b) real-time data, (c) static data, and (d) the integrated frame containing 3D image data, real-time data and static data.

2.2. Requesting and answering principles of the device data

To ensure the quality and efficiency of 3D display, especially considering the 3D scenes needed to be loaded via Internet, the data is requested and answered hierarchically in the proposed system. When displaying, the whole network is illustrated firstly, then the device groups could be zoomed in, finally a random device in the group could be accessed. Such process is like browsing an electronic map [14]. Taking advantage of the hierarchical requesting and answering, not only the memory consumption of the server could be saved, but also the time delay on internet could be effectively reduced [15].

We demonstrate a whole data center including thousands of devices layer by layer, and the hierarchical requesting and answering strategy is shown in Fig. 3. Fig. 3 (a) illustrates the overall system structure of a data center, in which users could observe the spatial locations of cabinets and monitor environmental status such as room temperature and humidity. Then, any random cabinet is able to be accessed and the device installations and settlements could be checked, as shown in Fig. 3 (b). Fig. 3 (c) presents the 3D image of a device which is installed in the cabinet. The device information such as its operating system, ID and memory situation could be known on this layer. Fig. 3 (d) shows the data ports of the device. Under such interface, one can read port IP, subnet mask, and communication objects. Such a layer based management is clear for the users and easy to locate.

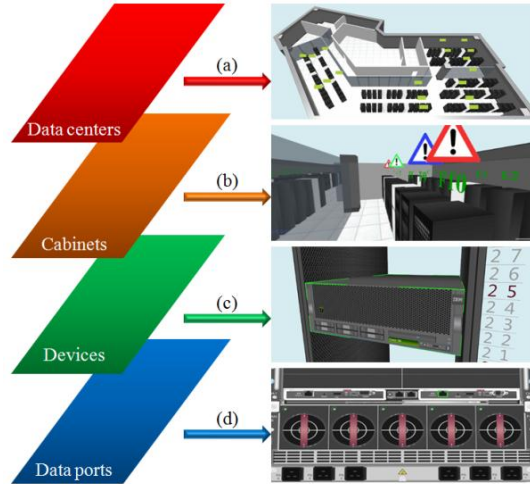


Fig. 3. Hierarchical 3D visualization for data management: (a) data center, (b) cabinets, (c) devices, and (d) ports.

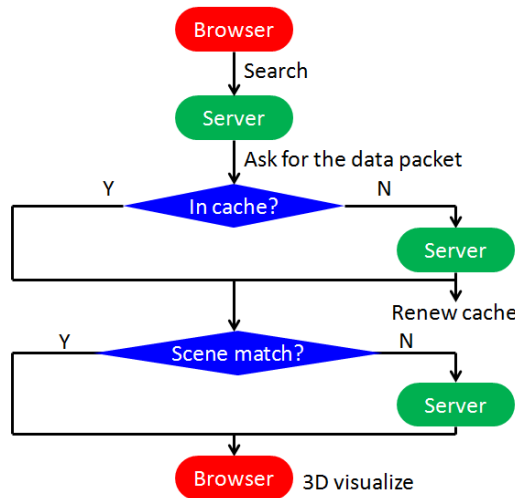


Fig. 4. Process for device location and information display.

2.3. The process of searching and displaying a device

The specific process of the device location and information display is shown in Fig. 4. First of all, the search request from the browser side is submitted to the remote server. Then a set of device data including only device IDs and device names is obtained from the server side. Such a requesting/answering process only takes several bytes to communicate. Then, the browser matches detailed information in data cache, according to the received IDs. If the information does not exist in cache, it would ask the server side for

the information, and renew the data cache. Otherwise, the browser judges if the 3D scene information matches the device, in order to make sure the device locate in the right place. The scene information is used to display the devices in a vivid and proper environment. If matches, 3D display would directly show on the browser (via data packet mentioned in Section 2.1). Otherwise, ask the server side for the scene information, and complete the process of 3D display.

3. Strategy of Multilevel Data Cache

To improve the communication efficiency of the 3D visualized data system, we design a multilevel cache in this paper, containing browser operation cache (BOC), browser file cache (BFC) and server operation cache (SOC). Among the 3 types of data cache, only BOC will be eliminated automatically when the browser is closed. The work flow of multilevel cache is shown in Fig. 5.

First of all, the data request is translated into a set of ID, and send to the BOC management module. If the needed data exists in BOC, the application in browser side would directly provide it back via data packet (see Section 2.1). Secondly, recorded the missed IDs (whose detailed information could not be found in BOC) and sent them to SOC management module in the server side in the form of XML. If the needed information have stored in SOC, send it to the browser side. Otherwise, send the missed IDs to SFC management module. In the third step, search corresponding data for the left ID set in BFC. If succeeded, read the matched data and return the information. Otherwise, send query for the left IDs to database and get the needed information. Finally, the application programs in the browser side analyze the data packets, and display the information on the screen to the users three dimensionally.

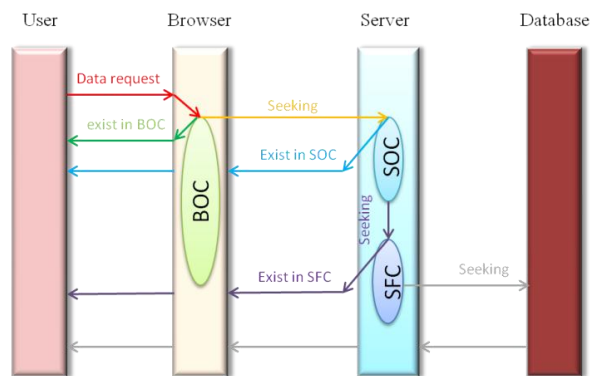


Fig. 5. Data processing in multilevel cache strategy.

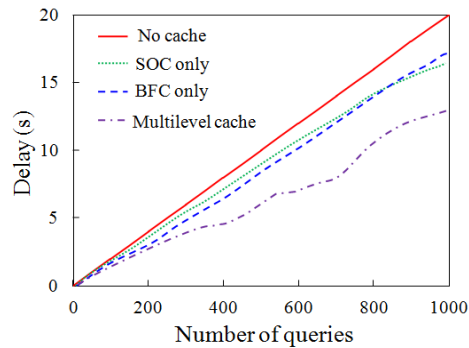


Fig. 6. Simulated correlation between the “number of queries” and “delay”.

Practically, there are quite a lot of queries from different users asking for the same data in the same device. For instance, users always pay more attention to the devices or ports which are usually erroneous. Thus, if excessive users submit queries to one port simultaneously, it may result network congestion, even

deadlock in the server. The strategy of multilevel cache can play an important role in balancing the workload of data querying between server and browser side, help to reduce the frequency database access and users' average waiting time, and improve the parallel computing ability. The advantages of the multilevel cache strategy could be simulated, as shown in Fig. 6. By setting 1000 devices, for 1000 queries asking for displaying randomly anyone of them (assuming that the probability to access any device is equal), the time delays for the "No cache", "BFC only", "SOC only" and "Multilevel cache" are presented respectively. It could be clearly observed that by utilizing the "Multilevel cache" strategy, the system communication efficiency would be dramatically promoted, depending on the network speed and cache capacity.

4. Experimental Analysis

To verify the advantages of the multilevel cache strategy, we measured the time delays to render 0~1000 cabinets (every cabinet contains 5 devices), and the time delays to render 50 cabinets for 0~20 clients simultaneously. By using the platform of the Flash Builder and MyEclipse, the experimental results are shown in Fig. 7 (a) and Fig. 7 (b) respectively. In Fig. 7, the red diamonds, blue cubs, green dots and purple triangles present the situations "No cache", "BFC only", "SOC only" and "multilevel cache" respectively. According to Fig. 7 (a), when there is no cache utilized, as cabinet number increases from 0 to 1000, the time delay to display them on screen increases from 4.3 s to 24.5 s. However, either using BFC or SOC, the time could be saved. Thus, by adopting multilevel cache with BFC, BOC and SOC, the time delay for displaying 1000 cabinets could be only 14.2 s. Compared to "No cache", such a data management has ~2 times higher efficiency. Moreover, based on the correlation between the simultaneous visitor number and the total delay in Fig. 6 (b), it is clear that the "multilevel cache" works more effectively, with 9.2 s for 20 visitors. For comparison, the delays for "No cache", "SOC only" and "BFC only" are 18.8s, 16.5s and 14.8s. In experiment, the initial delay is not 0, due to the software loading. In conclusion, the "multilevel cache" helps the 3D display speed reach ~0.005 s per device for one user, and ~0.04 s per device for 20 users simultaneously. Such experimental results verify the simulated results well, and this cache optimization also could be widely used to promote the efficiency of visualizations for many other applications [16]-[17].

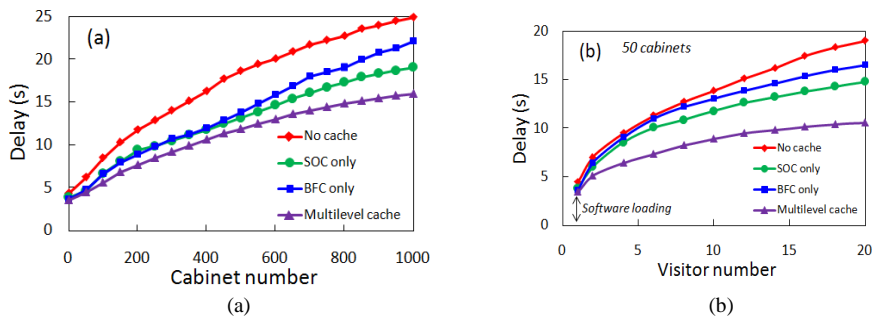


Fig. 7. Measured time delay: (a) Correlation of the cabinet number and the delay and (b) correlation of the visitor number and the delay.

5. Conclusions

This paper proposes an optimized 3D visualization scheme based on multilevel cache, which could be used as an efficient tool for device information management. Utilizing hierarchical data access method, we provide an effective method to visualize data networks, cabinets and devices just via Browser/Server system. Moreover, adopting multilevel cache strategy, the response delay of the visualization could be 50% lower than conventional method without cache. Such techniques may provide not only an efficient approach for the device data managing, but also a helpful reference for other complex data analyzing and processing.

References

- [1] Gungor V, Sahin D, Kocak T, Ergut S, Buccella C, Cecati C, Hancke G. A survey on smart grid potential applications and communication requirements. *IEEE Transactions on Industrial Informatics*, 2013; 9(1):28-42.
- [2] Eckhardt J, Rantala J. The role of intelligent logistics centers in a multimodal and cost-effective transport system. *Procedia-Social and Behavioral Sciences*, 2012; 48:612-621
- [3] Wang F. Parallel control and management for intelligent transportation systems: concepts, architectures, and applications. *IEEE Transactions on Intelligent Transportation Systems*, 2010; 11(3):630-638.
- [4] Chen X, Sun S, Qian F. A fast power system network topology based on tracking technology. *Power Sys. Tech.*, 2004; 28(5): 22-24.
- [5] Lange B, Rodriguez N, Puech W, Rey H, Vansques X. A 3D particle visualization system for temperature management. *Visualization and Data Analysis, SPIE*, 2011.
- [6] Beloglazova A, Abawajyb J, and Buyyaa R. Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. *Future Generation Computer Systems*, 2012; 28:755-768.
- [7] Niyato D, Lu X, Wang W. Machine-to-machine communications for home energy management system in smart grid. *IEEE Communications Magazine*, 2011; 49(4):53-59.
- [8] Speitkamp B, Bichler M. A mathematical programming approach for server consolidation problems in virtualized data centers. *IEEE Transactions on Services Computing*, 2010; 3(4):266-278.
- [9] Zhai G, Zhu X. Design and implementation of equipment management system based on flow chart using SVG. *Computer Engineering and Design*, 2012; 33(3):1208-1212.
- [10] Goodwin S, Dykes J, Jones S, et al. Creative User-centered visualization design for energy analysts and modelers. *IEEE Transactions on Visualization and Computer Graphics*, 2013; 19(12):2516-2525.
- [11] Chen J, Chen Y, Du X, et al. Big data challenge: A data management perspective. *Frontiers of Computer Science*, 2013; 7(2): 157-164.
- [12] Brink L, Stoter J, Zlatanova S. Establishing a national standard for 3D topographic data compliant to City GML. *International Journal of Geographical Information Science*, 2013; 27:92-113.
- [13] Tu JF, Guo RF. The application research of mixed program structure based on client-server, browser-server and web service. In: *Proc. of IEEE International Conference on Business Management and Electronic Information (BMEI)*, 2011:193-195.
- [14] Wu Y J, Wang Y, Qian D. A Google-map-based arterial traffic information system. In: *Proc. Intelligent Transportation Systems Conference*, 2007:968-973.
- [15] Chen J, Xu J, Li M, Xie B. Multiscale data organization of 3D models in network environment. *Science of Surveying and Mapping*, 2011; 36(6):182-184.
- [16] Schnorra L, Huardb G, Navaux P. TRIVA: Interactive 3D visualization for performance analysis of parallel applications. *Future Generation Computer Systems*, 2010; 26(3):348-358.
- [17] Upson C, Faulhaber T, Kamins D, et al. The application visualization system: a computational environment for scientific visualization. *IEEE Comp. Graph. and Appl.*, Jul. 1989:30-42.