

# Improvement of Scalability in Sharing Visualcasting Contents for Heterogeneous Display Environments

---

**Arata Endo**<sup>\*†</sup>

*Osaka University*

*E-mail: [endou.arata@ais.cmc.osaka-u.ac.jp](mailto:endou.arata@ais.cmc.osaka-u.ac.jp)*

**Yoshiyuki Kido**

**Susumu Date**

**Yasuhiro Watashiba**

**Kiyoshi Kiyokawa**

**Haruo Takemura**

**Shinji Shimojo**

An e-Science infrastructure for efficient discussion between researchers at remote sites is required. A Tiled Display Wall (TDW) can display visualization contents on a large, high-resolution display constructed from multiple monitors arranged in a matrix. To realize this infrastructure, visualization contents need to be shared between multiple TDWs at remote sites. Visualcasting is a technique that shares visualization contents between multiple TDWs which have heterogeneous display environments. However, Visualcasting has a scalability issue with the number of TDWs. To improve this issue, we propose a dynamic rearrangement mechanism for Visualcasting. In the evaluation, our proposed mechanism is shown to improve scalability.

*International Symposium on Grids and Clouds 2016*

*13-18 March 2016*

*Academia Sinica, Taipei, Taiwan*

---

\*Speaker.

†A footnote may follow.

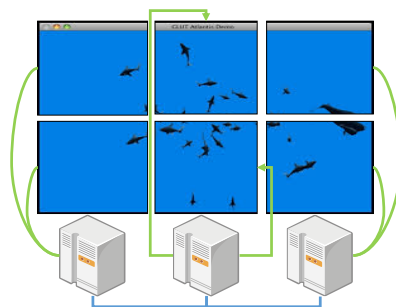
## 1. Introduction

In collaborative research, researchers require an infrastructure for efficient discussion between remote sites. Recently, collaborative research has increased across all disciplines. Moreover, in a growing number of cases researchers are transferred to other universities or institutions, and collaborative research is launched between the previous and currently affiliated universities and institutions. For collaborative research, researchers who belong to different universities and institutes need to communicate efficiently with each other.

In this situation, e-Science, meaning the set of tools and technologies for supporting collaborative networked sciences, has been received a lot of attention [7]. In the IEEE International Conference on e-Science [2], e-Science has been described as “e-Science promotes innovation in collaborative, computationally- or data-intensive research across all disciplines, throughout the research life cycle”. Most of the data generated from research is numerical. If the scale of the data is large, it is difficult to understand and analyze intuitively. Consequently, visualization is required to make the data easily understandable. Therefore, it is better to utilize visualization at the remote discussion because researchers can share the data generated from research in the discussion. For this reason, researchers require the e-Science infrastructure that shares visualization contents between remote sites and realizes efficient collaborative research.

A Tiled Display Wall (TDW) is appropriate for the e-Science infrastructure. TDW is a visualization device that consists of multiple displays arranged in a matrix. Multiple computers control these displays and create high-resolution and large-scale monitors in the TDW system. Figure 1 shows an example of a TDW system. The TDW has the following two advantages for efficient discussion.

1. The resolution of the TDW resolves a lack of information through the visualization contents that have high-resolution images and movies.
2. The scale of the TDW enables researchers to compare multiple applications on it simultaneously and to share their insights through discussion directly in front of the monitor.



**Figure 1:** An example of the TDW system

To realize the e-Science infrastructure, visualization contents must be shared between different TDWs. Visualcasting as proposed by Jeong [4] realizes this idea. Visualcasting is a technique that distributes visualization contents to multiple TDWs that have heterogeneous display environments.

All computers that control displays forming a TDW needs different image data because a computer renders different parts of an image in a display, and the configurations of TDWs are different between TDWs. Visualcasting supports the distribution of that part of the image corresponding to each computer.

However, Visualcasting does not have high scalability for the number of TDWs. In Visualcasting, a SAGE (Scalable Adaptive Graphics Environment) Bridge Cluster receives image data generated from applications and sends it to multiple TDWs. The SAGE Bridge Cluster has a static architecture that keeps the number of nodes in it static even if another TDW shares the visualization contents. Consequently, a lack of bandwidth is caused in the SAGE Bridge Cluster when many TDWs dynamically share the visualization contents.

In this research, taking into account the e-Science infrastructure where researchers in different sites can share the visualization contents efficiently, we tackle the scalability issue of Visualcasting. In Section 2, Visualcasting is explained and technical issues with this system are shown. In Section 3, our proposed method is described. Section 4 evaluates the proposed method. Finally, Section 5 presents our conclusion.

## **2. Visualcasting**

Jeong implements Visualcasting as a component of Scalable Adaptive Graphics Environment (SAGE) in [4]. SAGE is a middleware realizing a TDW system [5]. The component implemented for Visualcasting is named the SAGE Bridge.

### **2.1 SAGE**

SAGE is a TDW middleware developed by EVL (Electronic Visualization Laboratory) at the University of Illinois at Chicago [1]. SAGE allows each computer to launch a window of part of the TDW on the monitors. These windows construct an entire TDW and visualization contents are displayed on it by the cooperation of these computers.

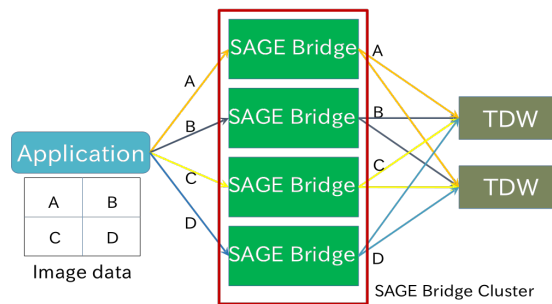
SAGE has three major features. First, SAGE can display the visualization contents generated in remote sites. This feature reduces the cost of setting the environment that runs applications. Second, a user can arrange multiple application windows on a TDW. Comparing multiple applications deepens the discussion. Finally, an application window can be shared between multiple TDWs. These features act effectively on the e-Science infrastructure.

### **2.2 SAGE Bridge**

SAGE Bridge is implemented in [4]. Moreover, it is confirmed in [8] that the SAGE Bridge can scalably support real-time distribution of extremely high-resolution images. Therefore, SAGE Bridge realizes Visualcasting.

An architecture of the SAGE Bridge is shown in Figure 2. The SAGE Bridge constructs SAGE Bridge Cluster for the distribution of image data delivery. SAGE Bridge Cluster contains one or more SAGE Bridge nodes that receive image data from applications and send the image data to each TDW. SAGE Bridge allows each TDW to have heterogeneous environments.

We explain the behavior of the SAGE Bridge. First, an application generates an image and divides it into sub-images. Second, the application sends the sub-images to the SAGE Bridge



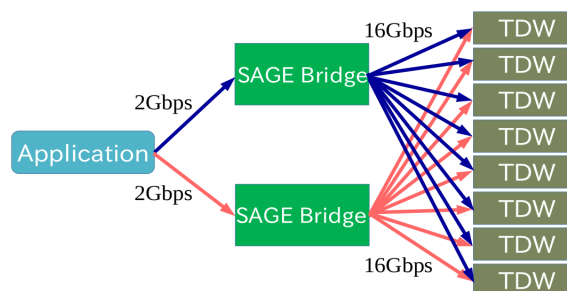
**Figure 2:** An architecture of SAGE Bridge

Cluster. Third, each SAGE Bridge node receives part of the sub-images and sends them to TDWs. Finally, each TDW receives all sub-images and reconstructs the original image from them.

### 2.3 Technical issue

Visualcasting has a scalability issue. When the number of TDWs is increased greatly, a lack of bandwidth is caused in each SAGE Bridge node. The number of SAGE Bridge nodes is static and each bandwidth used by each node for transporting image data is limited. If the current number of TDWs is much greater than the limited number, the bandwidth is insufficient. Therefore, low scalability for the number of TDWs becomes a problem for Visualcasting.

Figure 3 shows an example of the scalability issue of Visualcasting. In the figure, 8 TDWs share an application window and there are two SAGE Bridge nodes, each with a bandwidth of 10 Gbps. When the application generates image data in 4 Gbps and sends this data to each SAGE Bridge node in 2 Gbps, each SAGE Bridge node sends this data to each TDW in 2 Gbps. Therefore, each SAGE Bridge node needs the bandwidth of 16 Gbps which is greater than the bandwidth of 10 Gbps, and thus, presents a problem.



**Figure 3:** The Visualcasting scalability problem

Visualcasting scalability problem is a problem of the e-Science infrastructure. In other words, the lack of bandwidth causes a lack of data, and a delay of data transmission. This lack of data and the delay make discussions inefficient. Therefore, we propose a mechanism to improve the scalability issue of Visualcasting in this research.

### 3. The proposed mechanism

In this research, we propose and implement a dynamic rearrangement mechanism of the SAGE Bridge for the number of TDWs based on the original SAGE Bridge. The mechanism and the prototype implementation of the SAGE Bridge is explained.

#### 3.1 The dynamic rearrangement mechanism

The architecture of the SAGE Bridge Cluster requires a dynamic rearrangement mechanism where SAGE Bridge nodes are added dynamically corresponding to the load of bandwidth. In the conventional SAGE Bridge Cluster, the load of bandwidth is distributed to each SAGE Bridge node. However, an increase in the number of TDWs increases the load of the bandwidth in each SAGE Bridge node because the number of SAGE Bridge nodes is static. Therefore, the SAGE Bridge must rearrange the load distribution of bandwidth using the new SAGE Bridge nodes that are added dynamically.

We propose a SAGE Bridge that contains the dynamic rearrangement mechanism. An example of the proposed SAGE Bridge is shown in Figure 4. For the sake of simplicity, the divisions of image data are not executed and only one application is shared between multiple TDWs in this example. In this mechanism, all SAGE Bridge nodes are arranged hierarchically. A node receives image data from applications or from another node and sends the data to other nodes or TDWs. When a new TDW shares an application and a node connected to TDWs does not have enough bandwidth, a new node is added to the SAGE Bridge Cluster dynamically. The new node is inserted between TDWs and an existing node connected to TDWs and plays the role of relaying image data between them. In this way, our proposed mechanism rearranges the load distribution of the bandwidth dynamically.

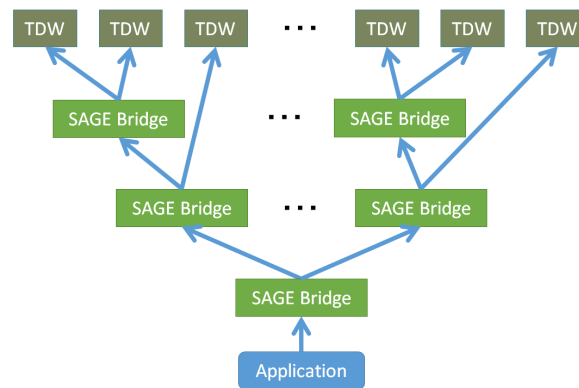
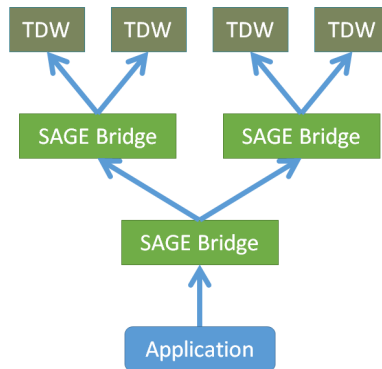


Figure 4: An example of the proposed SAGE Bridge

#### 3.2 Implementation of the prototype

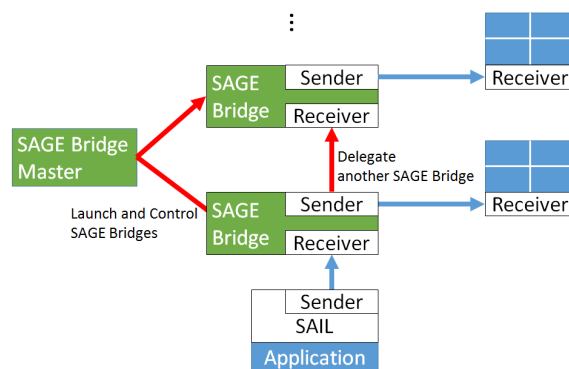
We implemented the prototype of the proposed SAGE Bridge based on the original SAGE Bridge. The prototype is shown in Figure 5. In this prototype, the architecture is a binary tree and the depth of the tree is shown as two for simplicity. Moreover, the process of adding a new

node is executed when the first and second TDWs share the application. This prototype realizes the dynamic rearrangement mechanism of the SAGE Bridge.



**Figure 5:** Prototype of the proposed SAGE Bridge

In the implementation, we rearranged the connection process of the SAGE Bridge. The prototype enables SAGE Bridge nodes to connect to one another because the SAGE Bridge nodes are arranged hierarchically in the proposed SAGE Bridge. Figure 6 shows a component of conventional the SAGE Bridge. The SAGE Bridge has a master component that controls the SAGE Bridge nodes and slave components that relays an image data between applications and TDWs as the SAGE Bridge node. When the SAGE Bridge is launched, a master is launched. The master launches multiple slaves in each computer as decided by the administrator of SAGE Bridge. The slave has a sender component that sends the image data and receiver component to receive image data. Our modification allows the SAGE Bridge nodes to delegate the image data to another SAGE Bridge node. The application also has a sender and TDW has a receiver. In the SAGE Bridge, the master receives system messages from applications and TDWs and sends them to slaves. Moreover, the slave decides the destination of the image data from the system messages. Therefore, we have implemented a connection process between the SAGE Bridge nodes in the master and slave components for the prototype.



**Figure 6:** A component of the conventional SAGE Bridge

In the connection process, we have added the dynamic addition of the SAGE Bridge nodes. When the master is launched, it launches no slaves in the proposed SAGE Bridge. When TDWs

share the application, the master launches slaves. Moreover, in the connection process, we added a process where a slave connects to other slaves with the same protocol that is used for the connections between a SAGE Bridge node and an application. The reason why it is possible to use the same protocol is that the connection protocol between a slave and an application and that of the connection between a slave and a TDW are the same. The architecture of the prototype for each number of TDWs was hard-coded. In this way, we implemented the prototype.

## 4. Evaluation

In this section, to validate that our proposed SAGE Bridge improves the scalability of Visual-casting, we compared the proposed SAGE Bridge and the conventional SAGE Bridge.

### 4.1 Evaluation environment

The experimental environment for the evaluation is shown in Figure 7. Each computer is interconnected via a switch. The paths between each node and the switch have approximately 1 Gbps bandwidth.

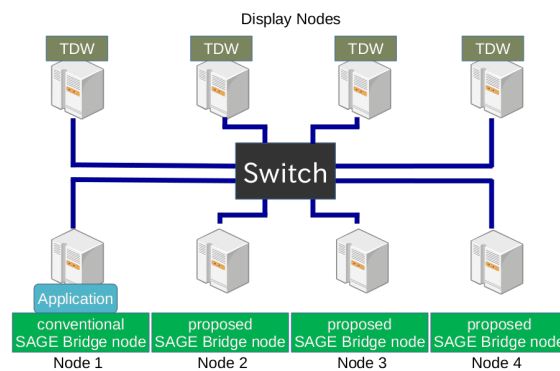


Figure 7: The evaluation environment

In Node 1, an application and the conventional SAGE Bridge node are executed. The application sends image data to the conventional SAGE Bridge node. The conventional SAGE Bridge node then sends the data to each TDW. In Nodes 2, 3 and 4, the proposed SAGE Bridge node is executed. These nodes relay the image data from the application of Node 1 to each TDW. In the Display Nodes, the TDWs are launched. The TDWs receive the image data from either the conventional or proposed SAGE Bridge nodes.

### 4.2 Evaluation method

In this evaluation, we validate that the proposed SAGE Bridge improves the load of bandwidth. For this reason, we compare the differences in bandwidth between the proposed and the conventional SAGE Bridges. If the proposed SAGE Bridge avoids a lack of bandwidth and the conventional SAGE Bridge causes the lack of bandwidth under the same conditions, this difference shows that the proposed SAGE Bridge improves the load distribution of the bandwidth.

For the comparison, we measured the bandwidth used for the transmission of the image data. In detail, we measure the bandwidth of each Display Node for each number of TDWs because a lower bandwidth in the Display Node than in the bandwidth required for transmission of the image data indicates that a lack of bandwidth occurs in SAGE Bridge nodes. The result of this measurement is used for the evaluation.

For the measurement, two experiments were done. One used the conventional SAGE Bridge, defined as Experiment A. The other used the proposed SAGE Bridge, defined as Experiment B. The steps for both experiments include the following:

1. launch four TDWs and the proposed or the conventional SAGE Bridge
2. launch an application and the application send image data to one of TDWs via the SAGE Bridge Cluster
3. measure the bandwidth in the Display Node where the TDW is executed
4. start to share the application window with another TDW
5. measure the bandwidth in each Display Node where the TDW shares the application window
6. steps 4 and 5 are repeated until the number of TDWs that share the application window is four

In the measurement, we use *Tcpstat* [3], which is a tool for measuring the bandwidth.

### 4.3 Evaluation result

We explain the result of both experiments. Tables 1 and 2 show the result of the measurement of the bandwidth in both experiments. Figure 8 is a graph that indicates the difference of the bandwidth between both SAGE Bridges. In Experiment A, the bandwidth decreases as the number of TDWs increases. This shows that a lack of bandwidth occurs in the SAGE Bridge Cluster. The reason why this lack occurs is that the bandwidth required for the transmission of the image data in a SAGE Bridge node is over 1 Gbps. In Experiment B, the bandwidth is kept high as the number of TDWs increases. When the number of TDWs is two and three, the graph should be flatter. However, the graph decreases a little because of bugs in the program of the proposed SAGE Bridge. Therefore, the result of Experiment B shows that the bandwidth is sufficient in each SAGE Bridge node.

**Table 1:** The measured bandwidth (Mbps) in Experiment A

#TDW	TDW1	TDW2	TDW3	TDW4
1	400	0	0	0
2	400	400	0	0
3	400	400	160	0
4	285	285	110	285



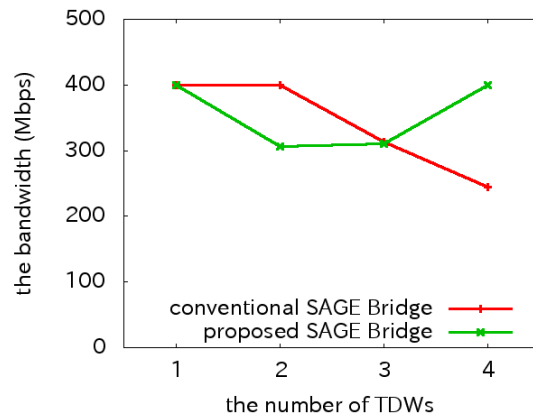


Figure 8: The evaluation result

A comparison of the results shows that the proposed SAGE Bridge avoids a concentration of bandwidth to a SAGE Bridge node through its dynamic rearrangement. This shows that the proposed SAGE Bridge improves the load distribution of bandwidth in Visualcasting. Therefore, the scalability issue of Visualcasting is improved.

## 5. Conclusion

The e-Science infrastructure requires a system of sharing visualization contents. Visualcasting is a technique that realizes this system. However, Visualcasting has a scalability issue. Therefore, we proposed a dynamic rearrangement mechanism for Visualcasting. This proposed mechanism has been implemented and experiments for the evaluation have been done in this research. As a result, the proposed mechanism has improved the scalability issue of Visualcasting.

For the future work, we will follow SAGE2 in the e-Science infrastructure. Recently, EVL has developed SAGE2 in [9]. SAGE2 is a Web based service updated from SAGE. Although its architecture has been changed drastically from the original SAGE, the network functionality has not been improved yet. Therefore, SAGE2 needs to be improved just as SAGE was improved. One area of research uses OpenFlow to avoid congestion in SAGE [6]. We also will try to apply this research to SAGE2 and to contribute to e-Science in our future work.

Table 2: The measured bandwidth (Mbps) in Experiment B

#TDW	TDW1	TDW2	TDW3	TDW4
1	400	0	0	0
2	400	250	0	0
3	400	250	400	0
4	400	400	400	400

## Acknowledgements

This work was partially supported by JSPS KAKENHI Grant Number 24700295 and 26540053.

## References

- [1] *EVL: Electronic Visualization Laboratory*, <https://www.evl.uic.edu/index.php>.
- [2] *IEEE International Conference on eScience | a web site for the conference series*, <https://escience-conference.org/>.
- [3] P. Herman, *The tcpstat tool*, <http://www.frenchfries.net/paul/tcpstat/>.
- [4] B. Jeong, *Visualcasting: Scalable Real-time Image Distribution in Ultra-High Resolution Display Environments*, PhD thesis, University of Illinois at Chicago, 2009.
- [5] B. Jeong, L. Renambot, R. Jagodic, R. Singhm, J. Aguilera, A. Johnson, and J. Leigh, *High-Performance Dynamic Graphics Streaming for Scalable Adaptive Graphics Environment*, In *The ACM/IEEE SC 2006 Conference*, page 24. IEEE, 2006.
- [6] Y. Kido, K. Ichikawa, S. Date, Y. Watashiba, H. Abe, H. Yamanaka, E. Kawai, H. Takemura, and S. Shimojo, *SAGE-based Tiled Display Wall enhanced with dynamic routing functionality triggered by user interaction*, *Future Generation Computer Systems*, 56:303–314, 2016.
- [7] N. Loassau, S. Rahmsdorf, T. Hey, and J. Hey, *e-Science and its implications for the library community*, *Library Hi Tech*, 24(4):515–528, 2006.
- [8] L. Renambot, B. Jeong, H. Hur, A. Johnson, and J. Leigh, *Enabling high resolution collaborative visualization in display rich virtual organizations*, *Future Generation Computer Systems*, 25(2):161–168, 2009.
- [9] L. Renambot, T. Marrinan, J. Aurisano, A. Nishimoto, V. Mateevitsi, K. Bharadwaj, L. Long, A. Johnson, M. Brown, and J. Leigh, *SAGE2: A collaboration portal for scalable resolution*, *Future Generation Computer Systems*, 54:296–305, 2016.