Cybermedia Center

Cybermedia Center
Osaka University

Osaka University, Japan

SC18 BOOTH 1825

About Us：Cybermedia Center, Osaka University

As a resource provider of knowledge and technology derived from advanced researches conducted in Osaka University, the Cybermedia Center (CMC) offers support in the areas of large-scale computation, information communication, multimedia content and education. The center also works closely with educational and research organizations within Osaka University, as well as with industries and institutes outside the University. By sharing its resources and encouraging local communities to use its facilities for public lectures and other events, CMC has helped to create a more internationally-oriented IT society for the region.

Location Map

Location

University-Wide Services

Large-Scale Computer System, we provide a high-performance computing environment, consisting of the NEC SX-ACE supercomputer and PC clusters, to both the academic and industrial communities. Part of the overall computer system is provided, as a computational resource, to the national High-Performance Computing Infrastructure(HPCI).

Information Media Education Multimedia Language Education, we have implemented a consistent curriculum, from the basics of computer utilization to advanced subject matter, while the Computer Assisted Language Learning System supports foreign language learning and cross-cultural understanding in accordance with each individual's language-proficiency level.

Cybermedia Commons is an active learning space for students, exploiting a wide variety of the Cybermedia Center's information technology, to support student's active learning and research activities.

Cybermedia Commons

Digital Library provides academic information databases and remote access to electronic journals. It is equipped with multimedia terminals and public network jacks with an authentication system.

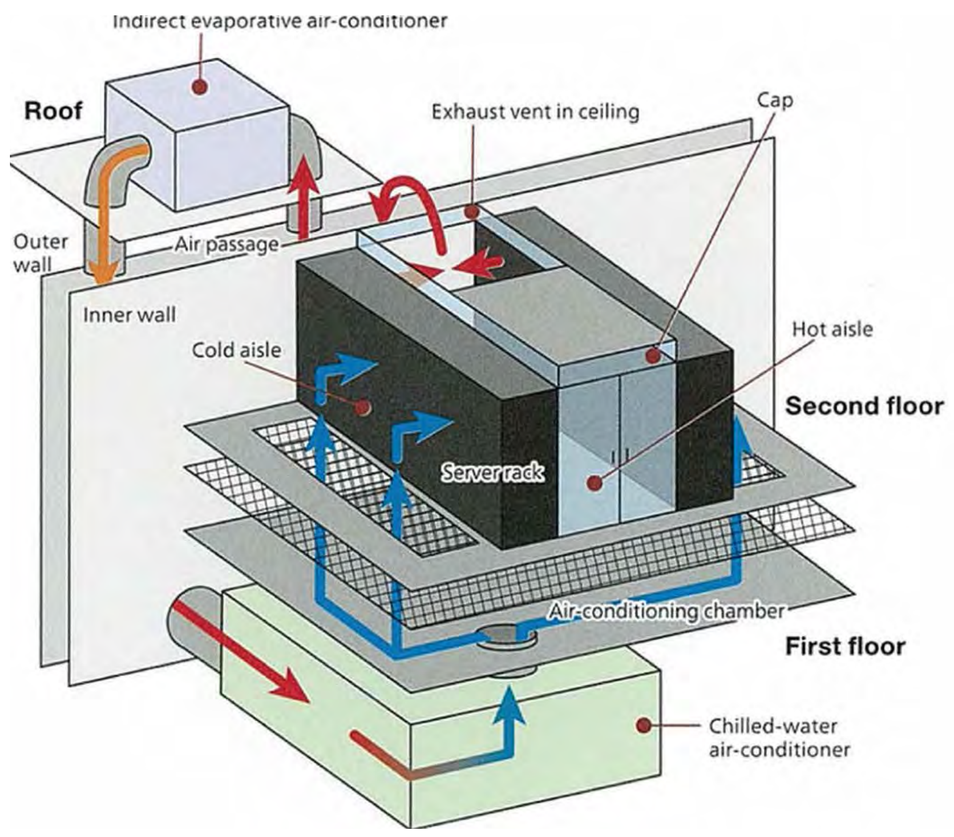
Repair and Maintenance of the Information Network, a high-speed, stable and reliable campus-wide network environment, as well as wireless access networks, as information infrastructure for supporting the educational, research, and social contribution activities of Osaka University.

Visualization Services, we maintain two types of high-resolution stereo visualization systems, as primary visualization facilities. The systems can be used for scientific visualization, information visualization, visual analytics, and other research activities.

Visualization Services

Academic Cloud improves the integration of computing resources scattered across the university. The objectives of the system are to optimize administrative operations, enhance security, and reduce costs.

IT Core Annex is a two-story steel-frame data center housing large-scale computers. The perimeter wall is designed with gently curving surface and light-permeable metal panels, to harmonize with the surrounding environment.



Air-conditioning mechanism in IT Core Annex

Contact : system@cmc.osaka-u.ac.jp Visit us at www.cmc.osaka-u.ac.jp

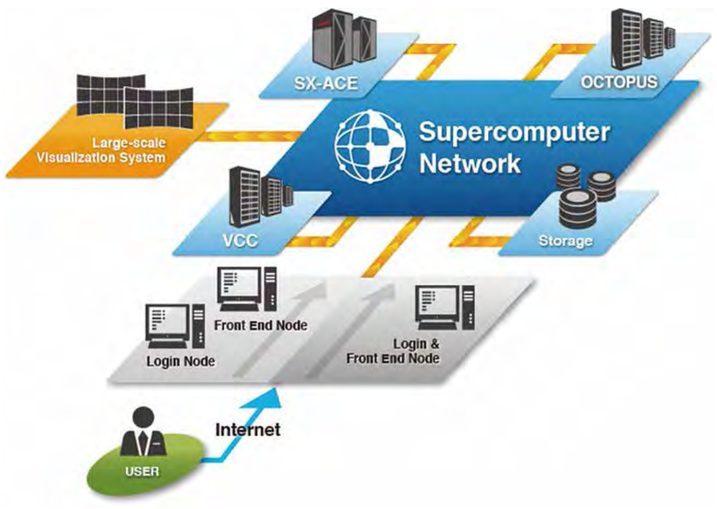
Cybermedia Center

Cybermedia Center
Osaka University

Osaka University, Japan

SC18 BOOTH 1825

Large-scale Computing and Visualization Systems at the Cybermedia Center



Overview of high-performance computing environment at the CMC

Large-scale computing systems (SX-ACE, VCC, and OCTOPUS), and large-scale visualization systems are deployed on CMC-Supercomputer network, a.k.a CMC-SCinet, a low-latency and wide-bandwidth network. This architectural design allows users to access to large-scale storage systems, perform large-scale high-performance computation and analysis on our large-scale computing systems, and then visualize its computation and analysis results without losing any important information on our large-scale visualization system.

Large-scale Computing System

The large-scale computing systems at the CMC are classified into (1) Vector-typed Supercomputer and (2) Scalar-typed Supercomputer.

SX-ACE

Type: Vector
OS: Super UX
# of nodes: 1536
# of cores: 6144
Peak performance: 423 TFlops

SX-ACE is a “clusterized” vector-typed supercomputer, composed of 3 cluster, each of which is composed of 512 nodes. Each node equips 4-core multi-core CPU and a 64 GB main memory. These 512 nodes are interconnected on a dedicated and specialized network switch, called IXS (Internode Crossbar Switch) and forms a cluster. Note that IXS interconnects 512 nodes with a single lane of 2-layer fat-tree structure and as a result exhibits 4 GB/s for each direction of input and output between nodes.

Library

MathKeisan(BLAS, LAPACK, etc)
ASL, ASLSTAT, ASLQUAD
MPI/SX
HPF/SX
XMP

Application

AVS/Express	TensorFlow	OpenFOAM
FreeFEM++	Torch	GAMESS
VisIt	Caffe	FLASH
Gaussian09/16	Theano	Octave
IDL	Chainer	Relion
LAMMPS	Quantum Espresso	GROMACS

OCTOPUS

Type: Scalar
OS: Linux
# of nodes: 319
Peak performance: 1.463 Pflops
Interconnect: InfiniBand EDR

OCTOPUS means Osaka university Cybermedia cenTer Over-Petascale Universal Supercomputer. OCTOPUS is a new cluster system supposed to start its operation in December 2017.This system is composed of different types of 4 cluster, General purpose CPU nodes, Xeon Phi nodes, GPU nodes and Large-scale shared-memory nodes, total 319 nodes. These nodes and large-scale storage “EXAScaler” are interconnected on InfiniBand EDR and form a cluster.

Library

Intel MKL(BLAS, LAPACK, etc)	IntelMPI, OpenMPI, MVAPICH2
ASL, ASLSTAT, ASLQUAD	XMP

General purpose CPU node × 236

CPU: Intel Xeon Gold 6126 × 2 (2.6 GHz, 12 cores)
Memory: 192 GB
Performance: 1.996 TFlops

Xeon Phi node × 44

CPU: Intel Xeon Phi 7210 (1.3 GHz, 64 cores)
Memory: 192 GB
Performance: 2.662 TFlops

GPU node × 37

CPU: Intel Xeon Gold 6126 × 2 (2.6 GHz, 12 cores)
Memory: 192 GB
Accelerator: NVIDIA Tesla P100×4
Performance: 23.196 TFlops

Large-scale shared-memory node × 2

CPU: Intel Xeon Platinum 8153 × 8 (2.0 GHz, 16 cores)
Memory: 6 TB
Performance: 8.192 TFlops

Storage

File system: DDN EXAScaler (Lustre)
Capasity: 3.1PB

VCC (PC Cluster for large-scale visualization)

Type: Scalar
OS: Linux
# of nodes: 69
Peak performance: 100.1 TFlops
Accelerator: NVIDIA Tesla K20×59

Library

Intel MKL(BLAS, LAPACK, etc)
Intel MPI, Open MPI

VCC is a cluster system composed of 69 nodes. These nodes are interconnected on InfiniBand FDR and form a cluster. Also, this system has introduced ExpEther, a system hardware virtualization technology. Each node can be connected with extension I/O nodes with which GPU resource, and SSD on 20Gbps ExpEther network. A major characteristic is that this cluster system is reconfigured based on user’s usage and purpose by changing the combination of node and extension I/O node.

Large-scale Visualization System

The large-scale visualization systems at the CMC are set up on Campus and on CMC’s Umekita Office. Large-scale and interactive visualization processing becomes possible through the dedicated use of PC cluster for large-scale visualization (VCC) on these systems. The visualization system in Campus is composed of 24 50-inch Full HD (1920x1080) stereo projection module (Barco OLS-521). Also, OptiTrackFlex13, a motion capturing system has been introduced in this visualization system. By making use of the software corresponding to the motion capturing system, interactive visualization leveraging Virtual Reality (VR) becomes possible.

Contact : system@cmc.osaka-u.ac.jp Visit us at www.cmc.osaka-u.ac.jp

Cybermedia Center

Cybermedia Center
Osaka University

Osaka University, Japan

SC18 BOOTH 1825

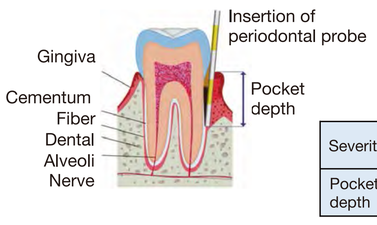
Secure and High Performance Data Analytics in HPC/Data Center

Data Analytics of Highly Confidential Medical Data

Example Application

Periodontal pocket depth estimation

- Periodontal disease or Pyorrhea may be a cause of diabetes and stroke. However, it is hard for people to notice the disease in the early stage because an apparent, subjective symptom rarely appears.
- This application facilitates regular self-check of the severity of periodontal disease and urges people to see dentists at the right timing.



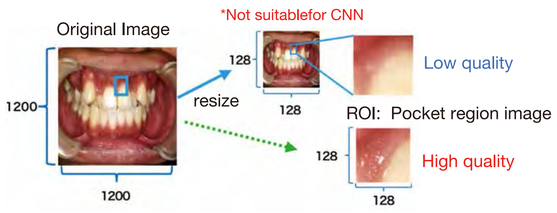
Severity	Healthy	Moderate Periodontitis	Severe Periodontitis
Pocket depth	≤ 3mm	4mm 5mm	≥ 6mm

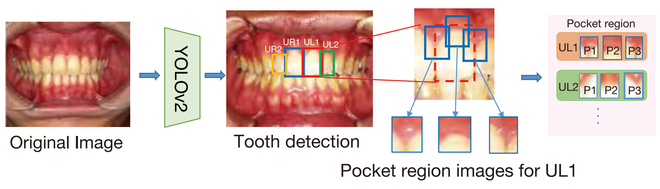
The proposed model overcomes two difficulties

Low quality of Region of Interest (ROI)

Problem:
Each pocket region is small in an oral image, the noise unrelated to the estimation is large.

Approach:
ROI feature extraction

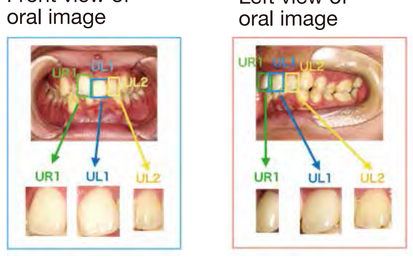




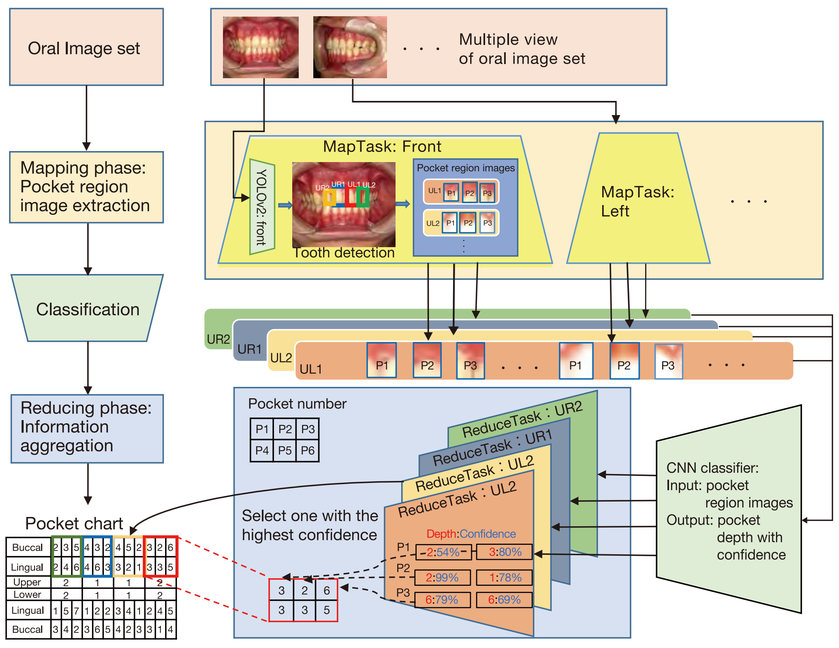
Teeth Heterogeneity

Problem:
The training dataset contains a set of oral images with different views. It is hard to recognize the identical teeth due to its shape, size, etc.

Approach:
Perform the pocket region extraction with respect to each view in parallel.



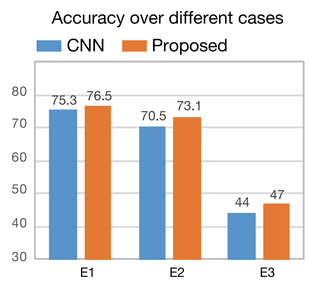
Design of the proposed MapReduce-like model

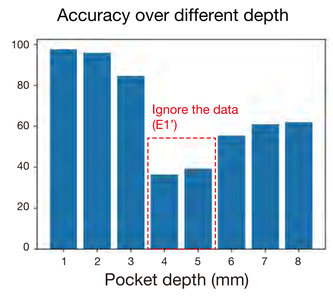


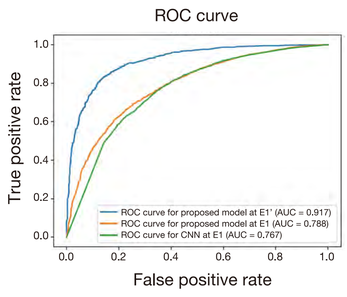
Evaluation

Experimental settings

E1: *Screening* by 2 stages, e.g., “Healthy” (3mm or less) and “Unhealthy” (4mm or more).
E2: *Severity measurement* by 3 stages, e.g., “Healthy” (3mm or less), “Moderate periodontitis” (4mm or 5mm), and “Severe periodontitis” (6mm or more).
E3: *Depth estimation* by 15 stages (pocket depth in millimeters between 1 and 15mm).





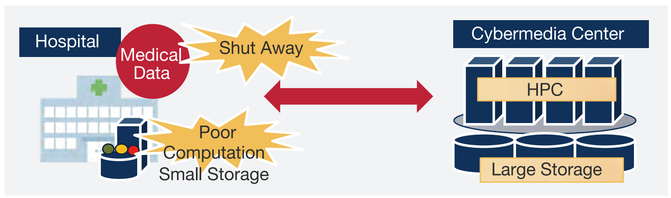


Dynamic Secure Staging for Processing Highly Confidential Data

Problem & Goal

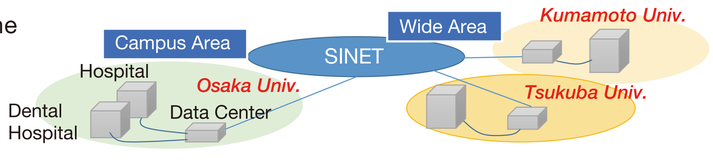
Problem Highly confidential data are forbidden to transfer.

- Location of data and processing power are different
- Collecting data from multiple location to make big data is necessary
- However, data transfer suffers from security issues



Goal Making use of confidential data in distributed location

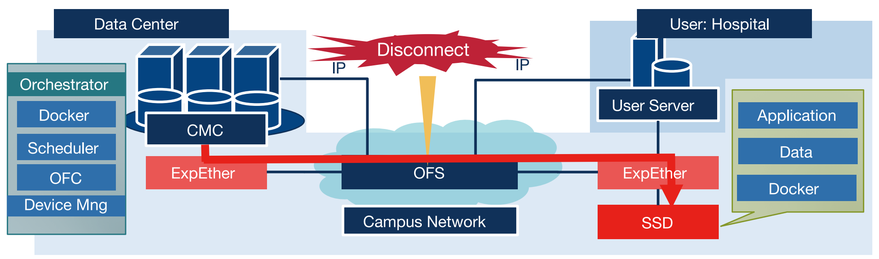
- Campus-scale via private line or campus LAN
- Multiple location via inter-university network



Different Location in Campus Area

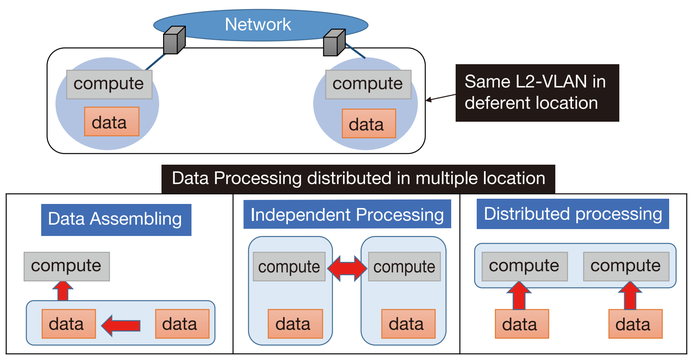
Managing security level by secure partitioning in multi-layer
- Data and resource are set in minimum partition and time slot

1. Open Flow Controller connects the server in the hospital to the DC.
2. SSD w/ data and Docker is PCIe-hot-plugged from hospital to DC machine
3. Scheduler run the Docker on the DC machine. Record result on the SSD.
4. SSD w/ data and result is PCIe-hot-plugged from DC to hospital machine
5. The hospital machine is dis-connected from the DC network



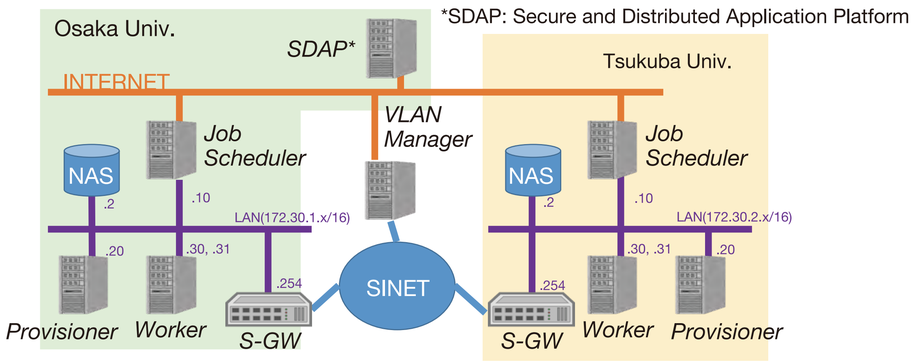
Different Location in Wide Area

- Not transfer chunk of raw data as much as possible
- L2-VLAN on Demand



Demonstration Setup

- Scheduler kicks resource provisioner w/ network controller.
- Distributed TensorFlow runs using workers and data on both location.



Y. Moriyama, C. Lee, K. Nozaki, Y. Kashiwagi, S. Murakami, Y. Matsui, A. Misawa, Y. Watashiba, S. Date, T. Yoshikawa, and S. Shimojo
Cybermedia Center & Graduate School of Dentistry, Osaka University: sc18@ais.cmc.osaka-u.ac.jp

Cybermedia Center
Osaka University

Cybermedia Center
Osaka University, Japan

SC18 BOOTH 1825

Access Control Based on Dynamic Network Management toward Connected-HPC

In the near future, data retrieved from IoT devices would be efficiently used and integrated in HPC simulations **(HPC x IoT)**. For the envisaged future, we have been developing an access control mechanism which an arbitrary set of IoT devices are dynamically connected to a HPC environment.

Background: Data Assimilation and IoT (Internet of Things) Era

What is Data Assimilation?

Data assimilation is a novel technique that combines observational data with numerical calculation for accurate and precise computation.

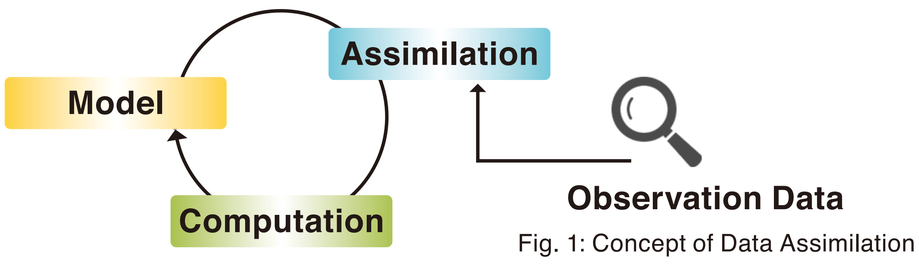


Fig. 1: Concept of Data Assimilation

It is predicted that the shorter the feeding interval of observation data to computation is, the higher the fineness of the result can be obtained. For the reason, IoT devices that capture observed data must be seamlessly integrated to HPC environments.

Sharing IoT Devices

In the IoT era, we envisage the world where an arbitrary set of IoT devices are used by an arbitrary group of users and applications. In other words, IoT devices are shared and used in an on-demand and sharing-economy fashion by an arbitrary group or community of users in the near future. For example, a set of cameras located in a lake may be used by a community of limnology scientists for a monthly period. On the other hand, the set of cameras may be used by some kind of weather monitoring application.

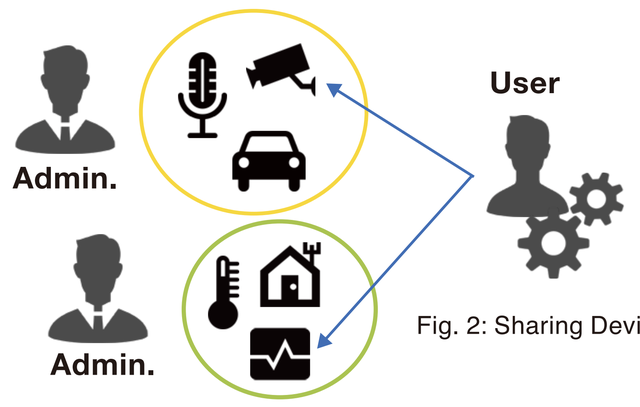


Fig. 2: Sharing Devices

Discussion: Real-time Data Assimilation and Issues

From the background described above, we work on the development of a new simulation technique that takes advantage of real-time observation data from an arbitrary set of IoT devices. We call this technique ***Real-time Data Assimilation (RDA)***. RDA can be applied to many scientific fields. To realize the simulation technique, **the HPC environment requires the connection to the outside network *(Connected-HPC)***. However, there are the following three technical issues to be tackled in today's HPC environments.

Fig. 3: Contributing to Many Fields by RDA

1. Packet reachability control is required from a security concern.

Today's HPC environment is usually a "closed" and "isolated" environment, meaning that HPC resources sit behind gateway servers connected to the external network. This situation is partly due to a fact that HPC administrators want to minimize the security risks from external networks. From this consideration, it would be helpful if we can connect and disconnect a HPC environment to the external network in an on-demand fashion.

2. Packet reachability should be controlled in a fine-grained fashion.

To minimize the security risks, the connect and disconnect of HPC resources to the external networks should be performed only when the access to IoT devices becomes necessary and unnecessary. For the reason, fine-grained control of network connection is preferred.

3. Network resources should be provided based on network administrators' policy.

Currently, a network is regarded as a shared "pipe". Taking the IoT era in the near future into consideration, an arbitrary IoT applications should be able to utilize resources of their interest including HPCs and networks based on their administrators' policy. From the idea above, a mechanism that can reflect users' needs and administrators' policy to network resources is essential.

Our Approach: Connected-HPC

For tacking the three technical issues above, we seamlessly integrate ***Software Defined Networking (SDN)*** and ***Job Scheduler*** (e.g. Slurm) for dynamic control of packet reachability between job and IoT devices for RDA technique.

To realize per-job packet reachability control as a fine-grained control of network connection, we have built a function that allows the scheduler to notify the SDN controller of the execution status of job. This notification contains the user ID on the HPC resource, and the IP address or ID of IoT devices to be connected/disconnected. Based on this notification, the SDN controller automatically searches the network path from HPC to the IoT devices and enables communication with those devices. **The function also enables the SDN controller to shut down the network path between HPC and IoT devices at the end of the job.**

Also, we have implemented **a function that the SDN controller can check whether the path adheres to the policy defined by the administrator and HPC user** before opening up the path between the HPC and the IoT devices. In other words, this check function makes it possible for the SDN controller to search for a path reflecting the policy of the stakeholders for each job.

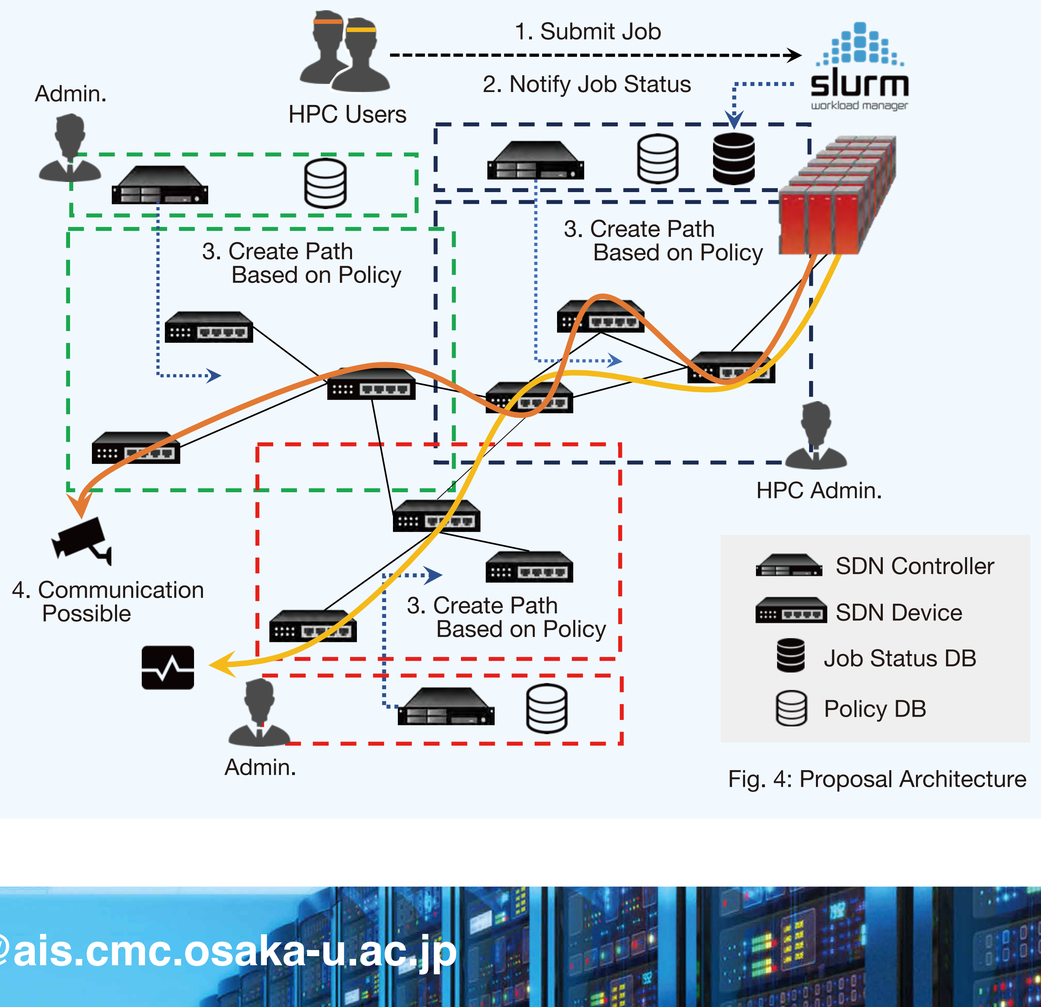


Fig. 4: Proposal Architecture

This work was partly supported by JSPS KAKENHI Grant Number JP17KT0083.

Hiroaki Morimoto: sc18@ais.cmc.osaka-u.ac.jp

Cybermedia Center

Cybermedia Center
Osaka University

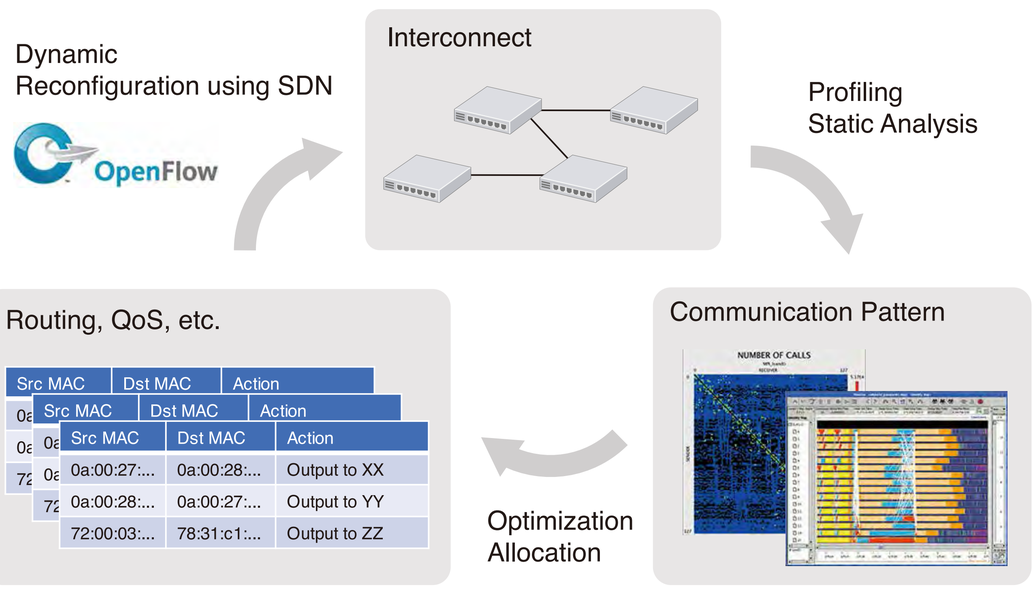
Osaka University, Japan

SC18 BOOTH 1825

Dynamically Optimized Interconnect Architecture Based on SDN

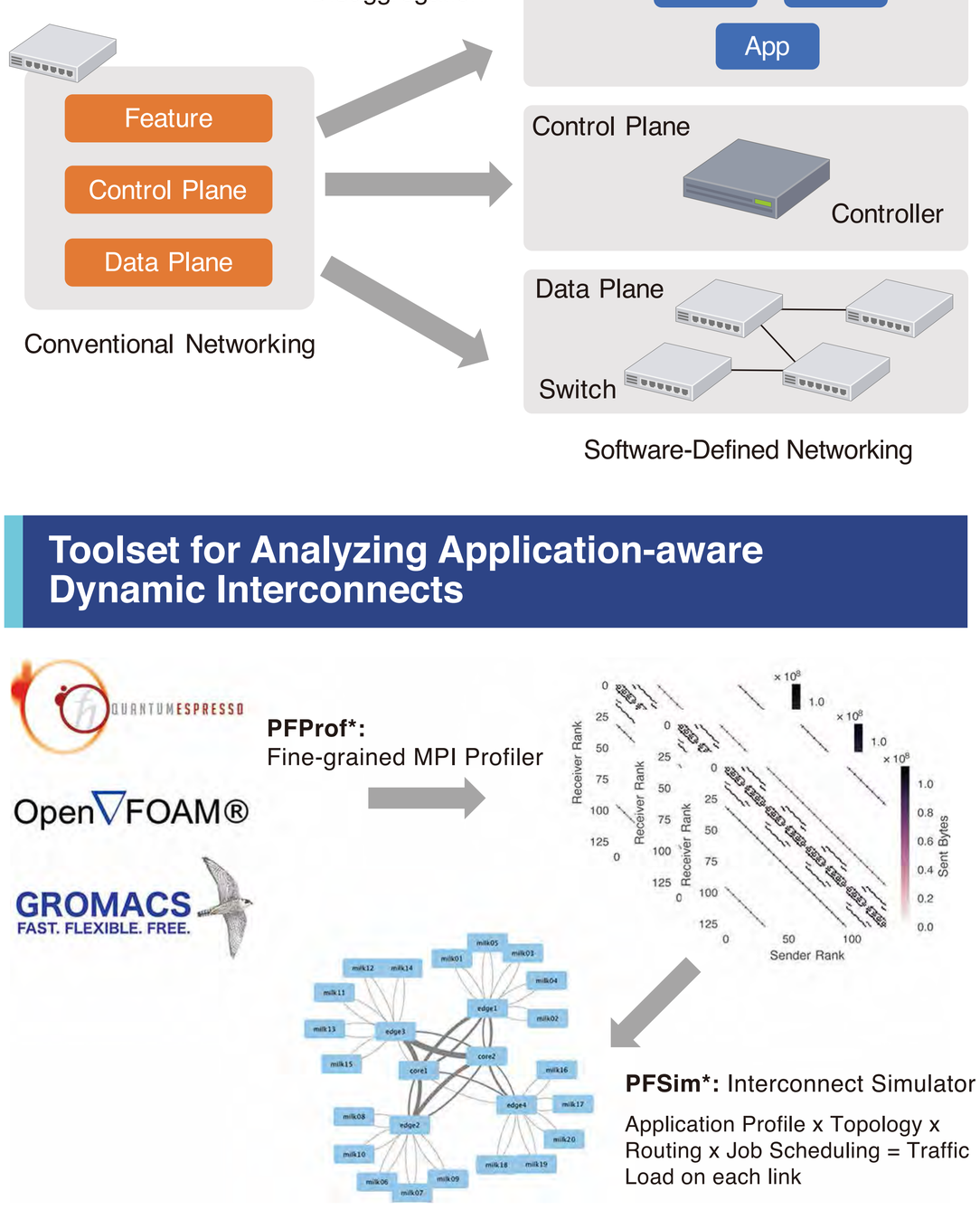
Fundamental Idea of SDN-enhanced MPI

Can we accelerate MPI communication and improve the utilization of interconnect by leveraging the network programmability of SDN?



What is Software-Defined Networking（SDN）?

Software-Defined Networking (SDN) is a novel network architecture that decouples conventional networking function into a programmable control plane (responsible for deciding how to control the packets) and a data plane (responsible for the actual packet delivery).



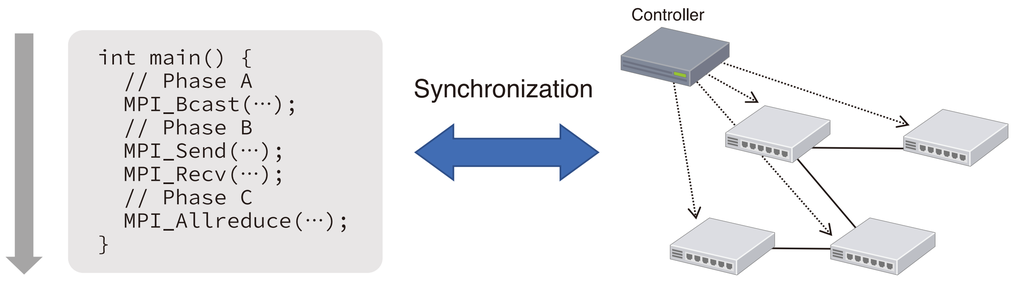
Toolset for Analyzing Application-aware Dynamic Interconnects

* Keichi Takahashi, Susumu Date, Khureltulga Dashdavaa, Yoshiyuki Kido, Shinji Shimojo, “PFAnalyzer: A Toolset for Analyzing Application-aware Dynamic Interconnects” , the Monitoring and Analysis for High Performance Computing Systems Plus Applications (HPCMASPA) Workshop, Cluster 2017, pp. 789-796, Honolulu, Hawaii, Sep. 2017.

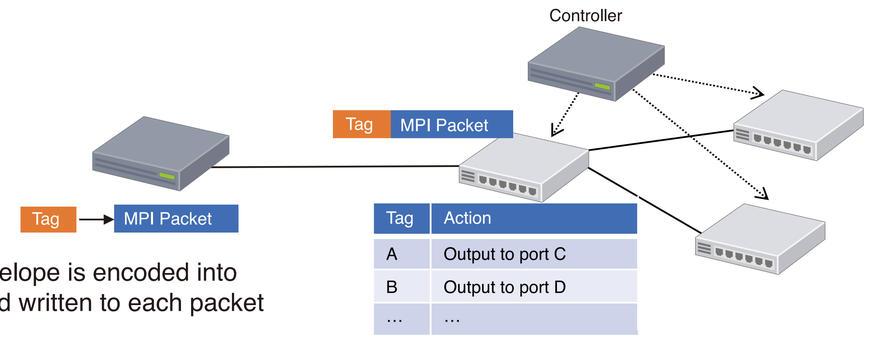
** Yohei Takigawa, Keichi Takahashi, Susumu Date, Yoshiyuki Kido, Shinji Shimojo, “A Traffic Simulator with Intra-node Parallelism for Designing High-performance Interconnects” , The 2018 International Conference on High Performance Computing & Simulation (HPCS 2018), July 2018.

Coordination Mechanism of Computation and Communication

How do we reconfigure the interconnect in accordance with the execution of application?



Encode the MPI envelope into a tag and embed into each packet in the kernel



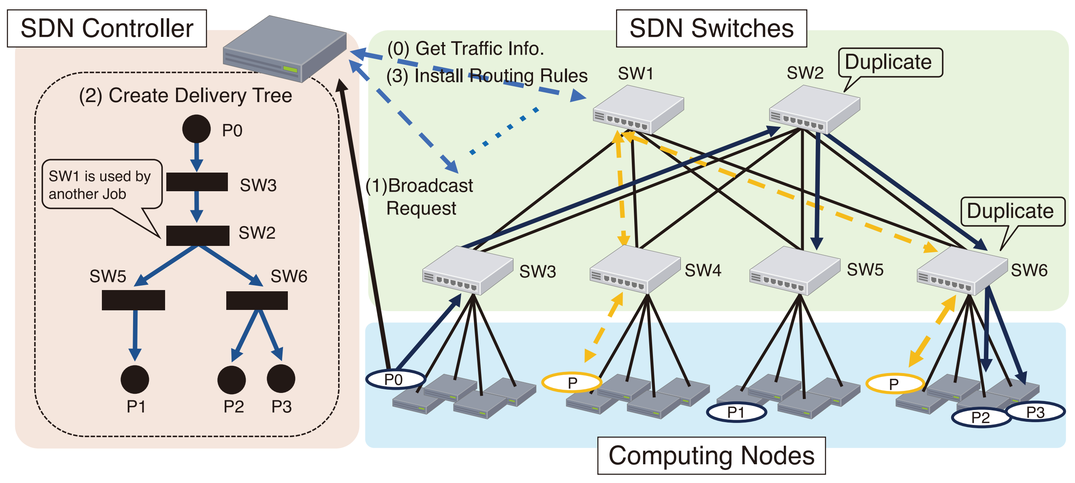
MPI envelope is encoded into a tag and written to each packet

Tag-based forwarding

Keichi Takahashi, Susumu Date, Dashdavaa Khureltulga, Yoshiyuki Kido, Hiroaki Yamanaka, Eiji Kawai, Shinji Shimojo, “UnisonFlow: A Software-Defined Coordination Mechanism for Message-Passing Communication and Computation” , IEEE Access, vol. 6, no. 1, pp. 23372-23382, 2018.

SDN-enhanced MPI Broadcast

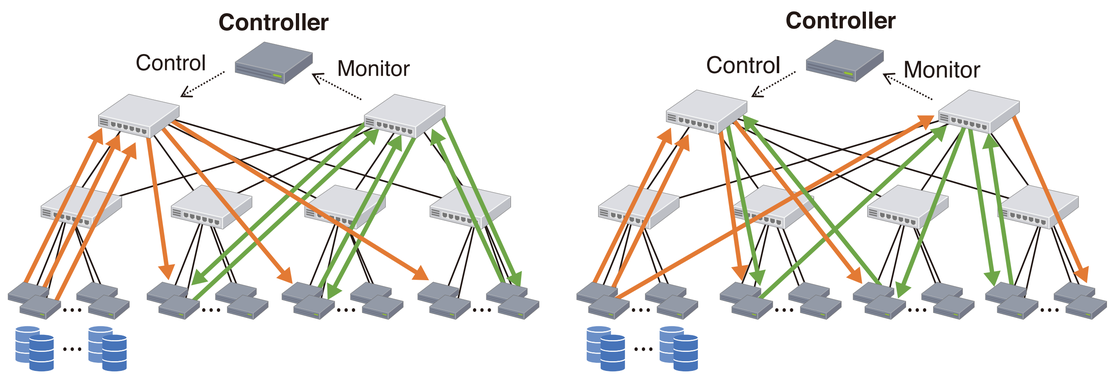
We propose an accelerated implementation of MPI broadcast using SDN (SDN-MPI_Bcast). In this implementation, the SDN controller dynamically installs broadcast route to SDN switches based on the topology of the interconnect and other jobs running in the same cluster.



Hiroaki Morimoto, Khureltulga Dashdavaa, Keichi Takahashi, Yoshiyuki Kido, Susumu Date, Shinji Shimojo, “Design and Implementation of SDN-enhanced MPI Broadcast Targeting a Fat-tree Interconnect” , The 2017 International Conference on High Performance Computing & Simulation (HPCS 2017), pp.252-258, Genoa, Italy, July 2017.

Contention Avoidance of Stage IO Communication and Inter-process Communication

We proposed two conflict avoidance methods to investigate whether the conflict between both types of communication has mutual influence on the performance of communication.



Link Sharing Conflict Avoidance

Link Separation Conflict Avoidance

Staging Communication

Inter-process Communication

Arata Endo, Ryoichi Jingai, Susumu Date, Yoshiyuki Kido, Shinji Shimojo, “Evaluation of SDN-based Conflict Avoidance between Data Staging and Inter-Process Communication” , The 2017 International Conference on High Performance Computing & Simulation (HPCS 2017), pp. 267-273, Genoa, Italy, July 2017.

These works were partly supported by JSPS KAKENHI Grant numbers JP16H02802 and JP17K00168.

Keichi Takahashi, Yohei Takigawa, Arata Endo, and Hiroaki Morimoto: sc18@ais.cmc.osaka-u.ac.jp

Cybermedia Center
Osaka University

Cybermedia Center

Osaka University, Japan

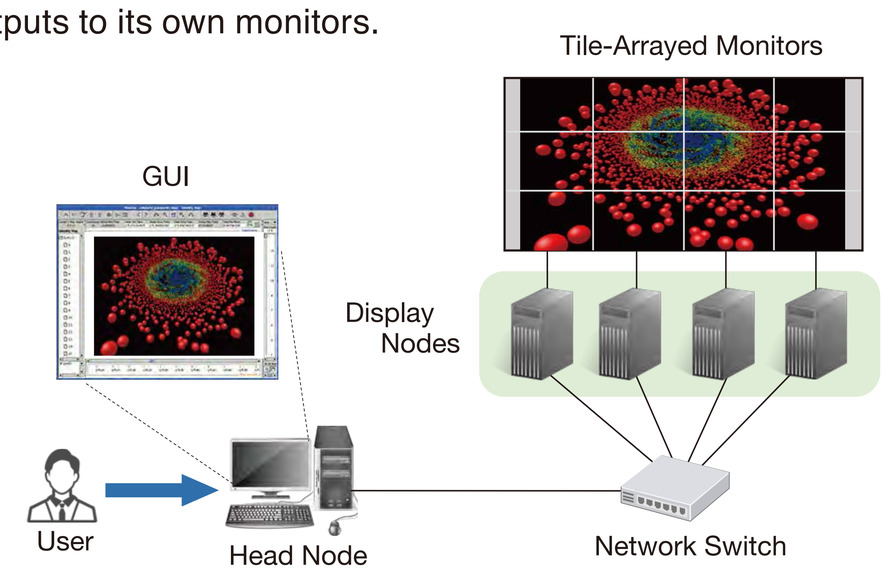
SC18 BOOTH 1825

Novel Mechanisms to Support Scientific Visualization on TDW

What is Tiled Display Wall (TDW)?

- A Tiled Display Wall (TDW) is a scalable visualization system, which can provide a virtual high-resolution screen **by combining multiple sets of computers and monitors.**
- A TDW is often **leveraged for scientific visualization.**
  - A TDW can visualize large quantities of scientific data without a lack of information. (*e.g. simulation results, network graph etc.*)
  - A lot of researchers can observe visualized data simultaneously and exchange ideas with each other on the spot.

- A general TDW has **a cluster-based architecture.**
  - The head node and the display nodes are cooperated by dedicated visualization software. (*e.g. SAGE2, ParaView, COVISE etc.*)
  - The head node provides a GUI of the TDW to users.
  - Each display node renders the fragment of visualized data and outputs to its own monitors.



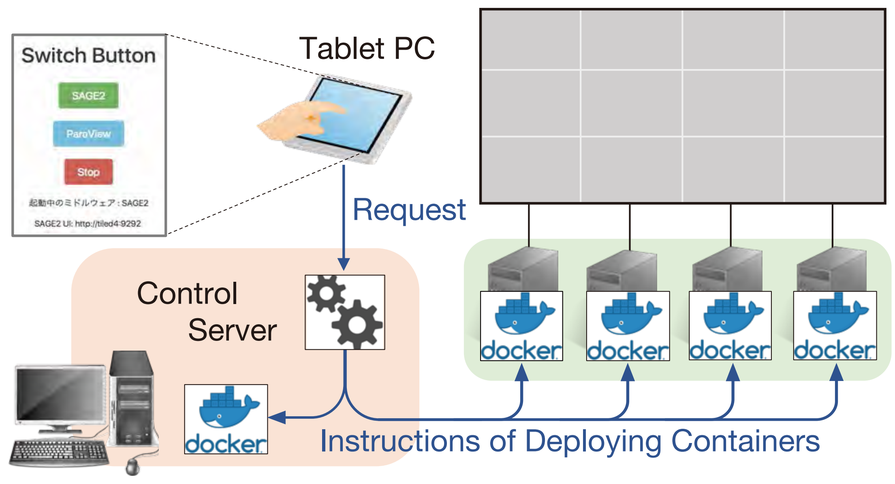
Switching Mechanism of Visualization Software for the Visualization Service

- The TDW in the Cybermedia Center (Osaka University) is leveraged by a lot of researchers as the Visualization Service.
- **Problem**: Frequent conflicts of dependent library versions
  - Researchers require to **install various visualization software** on the TDW in response to their analytic styles and data formats.
  - Most visualization software **depend on the particular versions** of system libraries and graphic libraries.
  - The administrators often **suffer from the complicated operations** for the conflict avoidance.

Software	Dependent Libraries
SAGE2 (v3.0.0)	Node.js (v10.0 or newer), FFmpeg (v3.0 or newer), ImageMagick (between v6.0 and v6.9)
ParaView (v5.6.0)	Qt (between v5.6 and v5.9), FFmpeg (v2.3), Python (v2.7)
COVISE (v2018.9)	OpenSceneGraph (v3.2 or newer), VTK (v6.0), Python (v3.5 or newer)

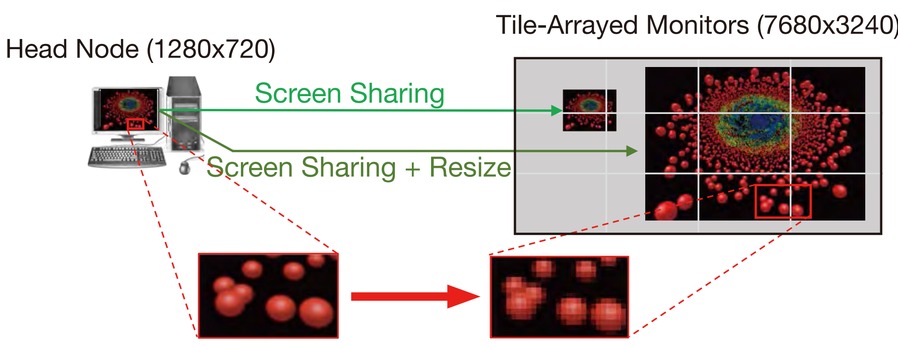
This work was supported by JSPS KAKENHI Grant Number JP26540053.

- **Our approach**: Switching mechanism using Docker
  - *Docker* separates the visualization software environments from each other with little overhead.
  - All the Docker containers are deployed **automatically in a single operation** on the tablet PC.

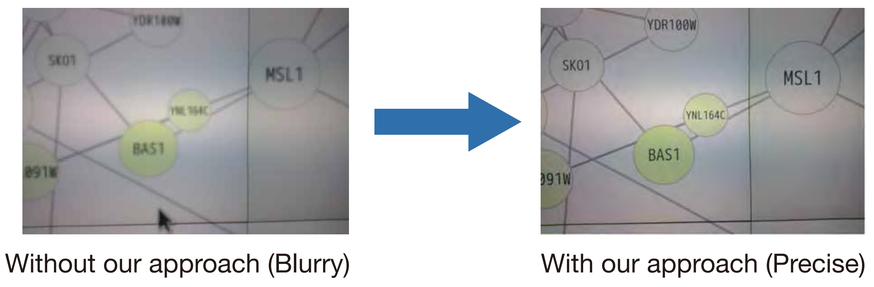


High-Resolution Streaming Functionality in SAGE2 Screen Sharing

- To use existing desktop applications on a TDW, users are required to modify the source code.
- *SAGE2* (popular visualization middleware) provides **Screen Sharing**, which is the function to stream user's desktop contents to a TDW.
  - Screen Sharing allows users to display a wide range of desktop applications on a TDW **without redevelopment.**
- **Problem**: Resolution constraint
  - The desktop contents are displayed at the **same resolution as the monitor of the head node.**
  - Large difference in the screen resolution will **deteriorate the visibility of desktop applications.**

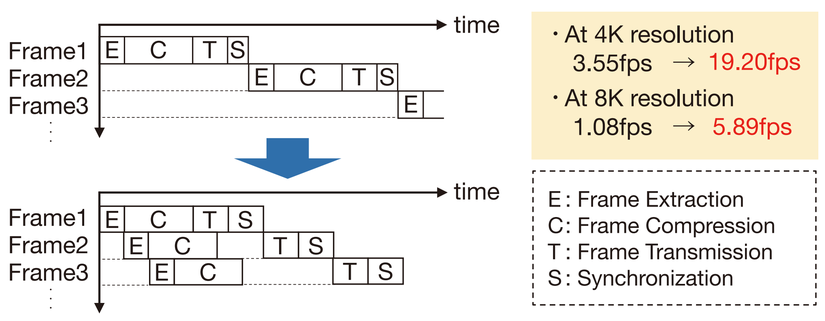


- **Our approach**: Xvnc and Pipeline streaming
  - *Xvnc* creates the virtual desktop screen **at an arbitrary resolution** on the head node regardless of the specifications of its monitor.



Without our approach (Blurry) With our approach (Precise)

  - To improve the frame rate in the high-resolution streaming, **the streaming process is pipelined.**



This work was supported by JSPS KAKENHI Grant Number JP18K11355 and "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" in Japan (Project ID: jh160056-ISH, jh170056-ISJ, jh180077-ISJ).

Kazuya Ishida, and Yoshiyuki Kido: sc18@ais.cmc.osaka-u.ac.jp