An overview of the storage and post-processing environment at RIKEN R-CCS

Jorji Nonaka and Yuichi Tsujita

{jorji, yuichi.tsujita}@riken.jp

Operations and Computer Technologies Division

RIKEN Center for Computational Science

Computer simulations create the future

HPC-IODC'19 (Jun. 20, 2019)

Outline

- **RIKEN R-CCS**
  - Operations and Computer Technologies Division
- **K computer environment**
  - Storage
    - File system configuration
  - Post-processing (visualization and analysis)
    - K pre/post processing server
    - K pre/post cloud system
    - K computer

Operations and Computer Technologies Division

- **RIKEN R-CCS Organization**

https://www.r-ccs.riken.jp/en/overview/organization

- **HPC Usability Development Unit**
  - Enhancement of the usability of the K computer
    - Post-processing environment
      - For K computer users
      - For internal members

K computer storage

- **Simulation Data**
- **Real World Data (Measured Data)**
- **Log Data**
  - System (Computational and Storage) and Facility

https://www.r-ccs.riken.jp/en/k-computer

Post-processing

- Post-hoc visualization and analysis

LFS

Simulation

Log Data

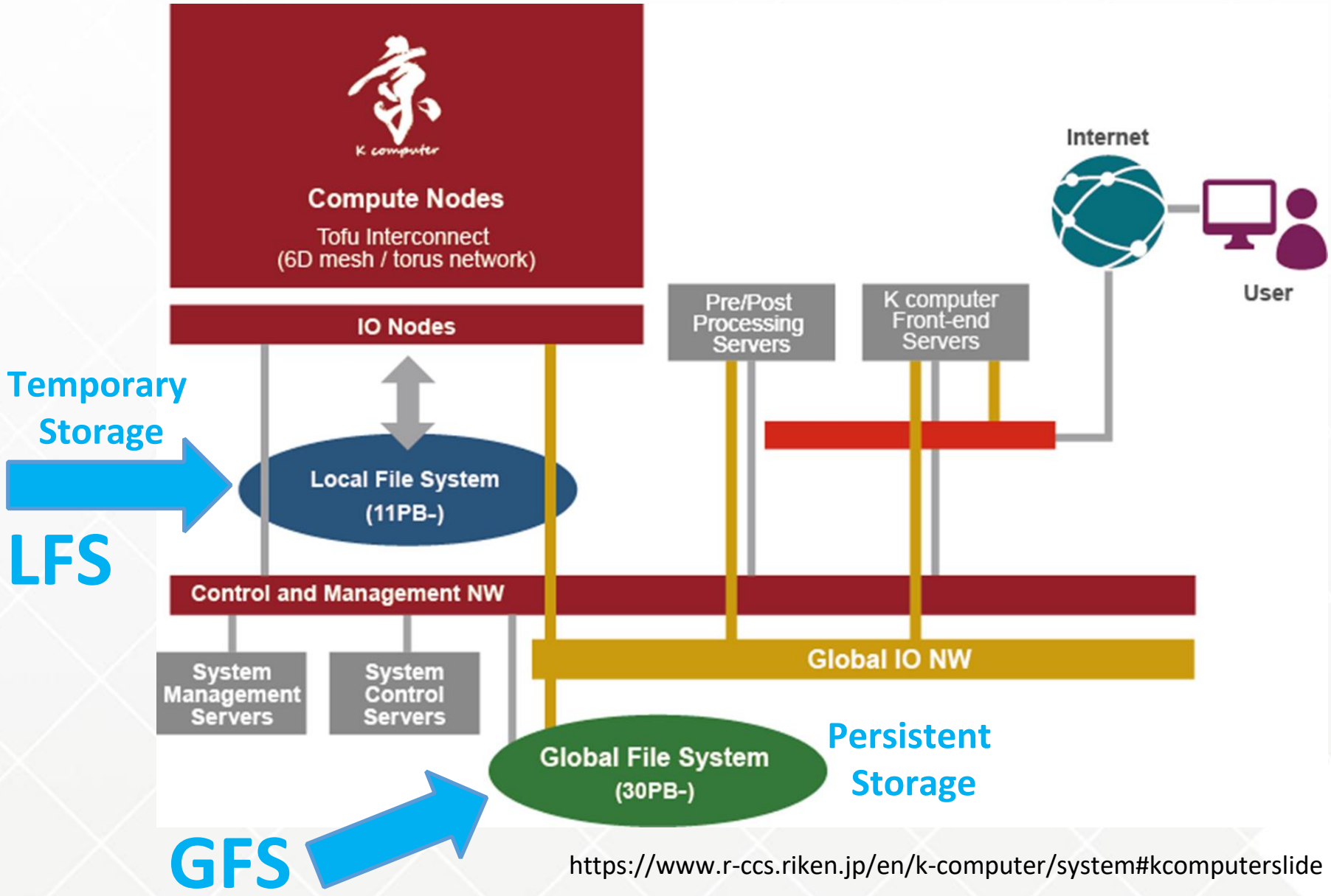
GFS

Visualization

Analysis

K computer environment

Computer simulations create the future

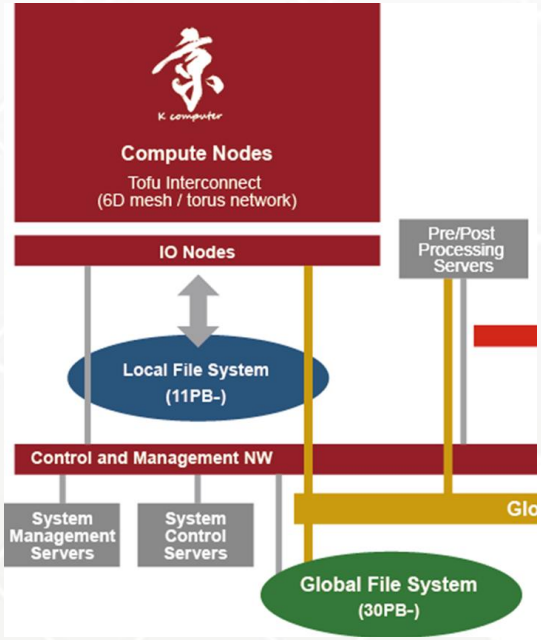


https://www.r-ccs.riken.jp/en/k-computer/system#kcomputerslide

File system configuration

- **Organization of file systems at the K computer**
  - **LFS** : Performance oriented
    - for high performance I/O during computation
  - **GFS** : Capacity oriented
    - for huge data storing and high redundancy

File system	LFS	GFS
Total volume size	~ 11 PB	> 30 PB
# volumes	1	8
# OSSs	2,592	90
# OSTs	5,184	2,880
Disk system of OST	RAID5	RAID6



Some OSS applications

Computer simulations create the future

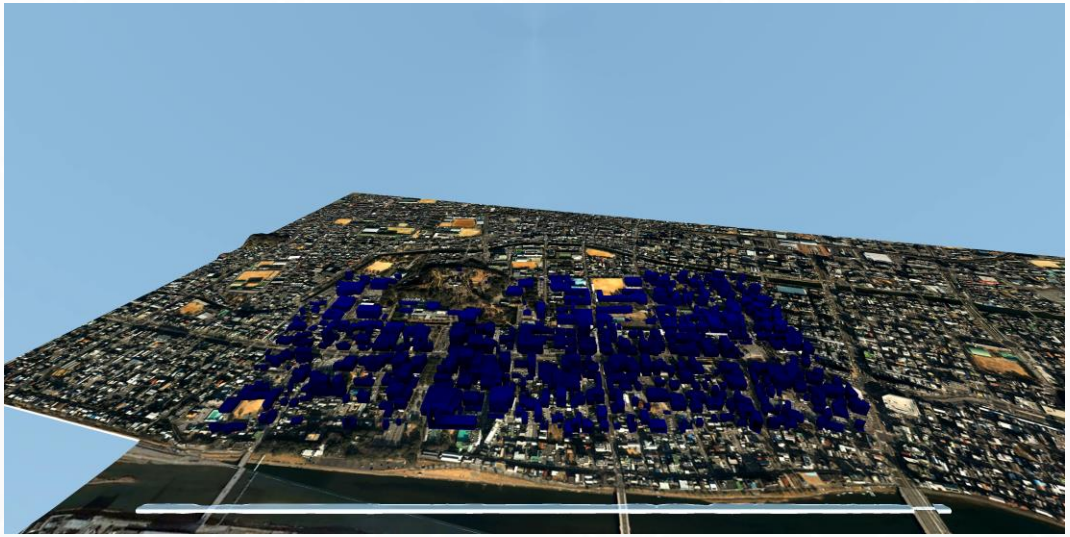
- **HIVE (R-CCS, Kyushu University)**
  - https://github.com/RIKEN-RCCS/HIVE
- **KVS (Kobe University)**
  - https://github.com/naohisas/KVS
- **PIDX/OpenVisus (University of Utah)**
  - https://github.com/sci-visus/PIDX
  - https://github.com/sci-visus/OpenVisus

Close relation with the developers

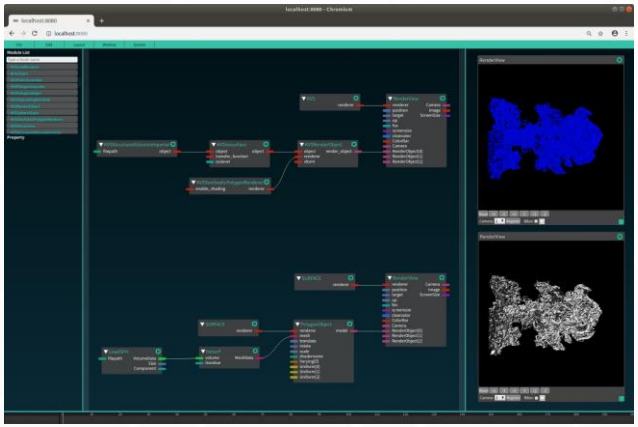
- **VTK-based Applications**
  - ParaView, VisIt
- **Other OSS applications**

Limited support to the users

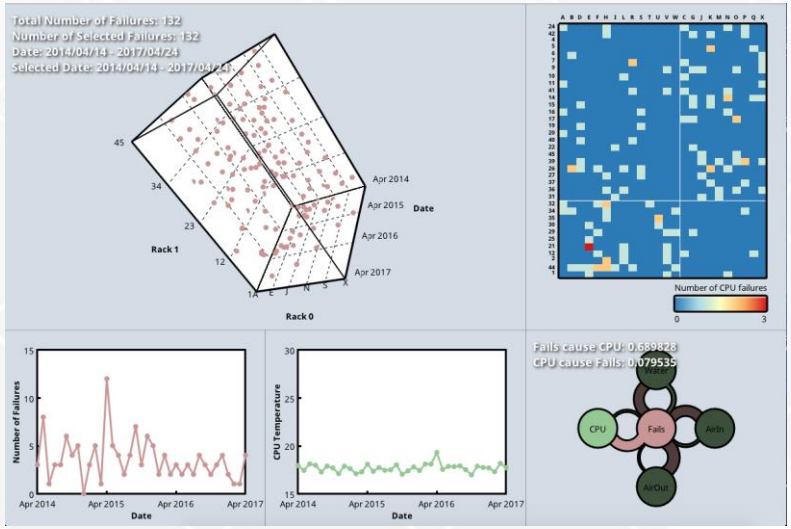
Some OSS applications



HIVE

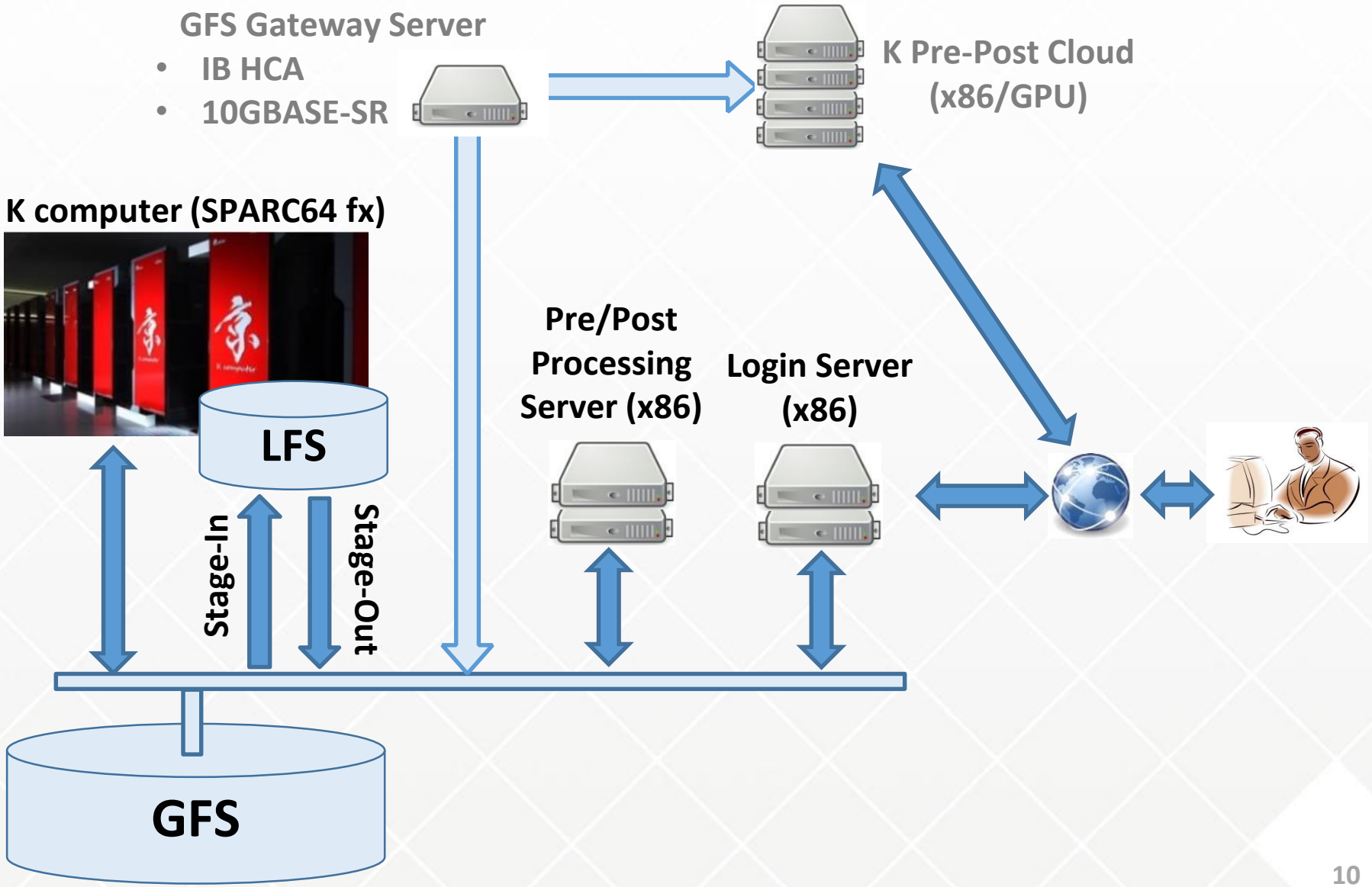


PIDX/OpenViSUS

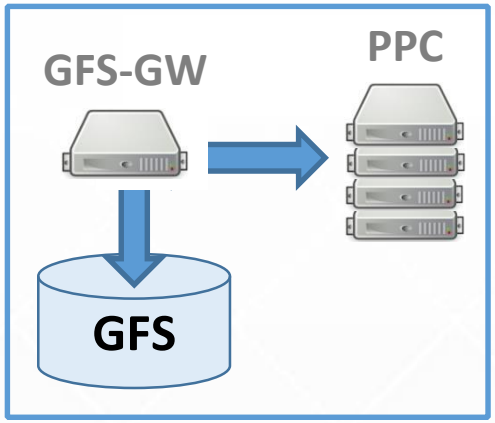


KVS

Post-processing environment

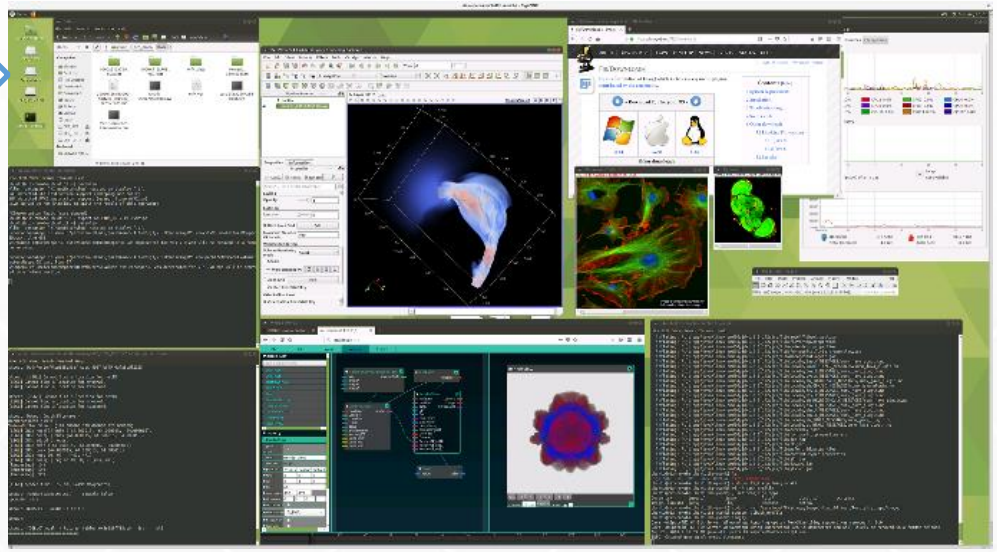


K pre-post cloud (GFS)



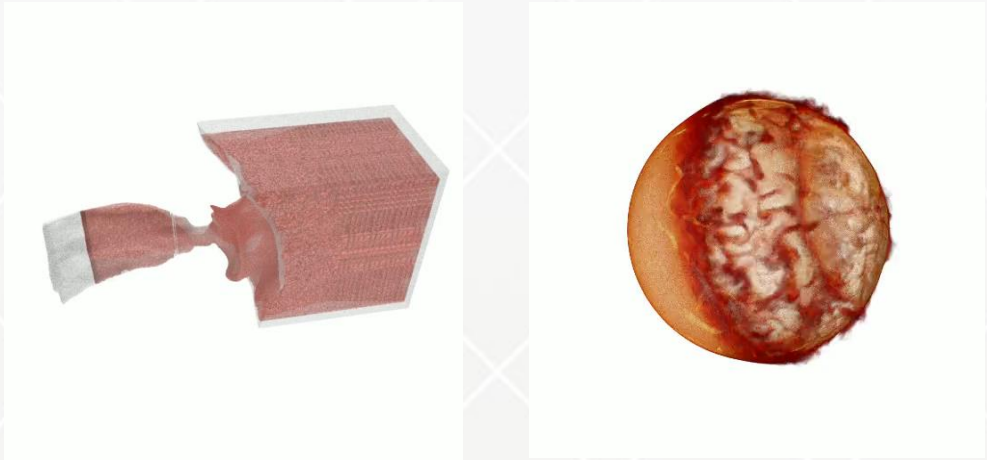
- **Interactive Visualization**

GFS mounted folders



Remote Visualization (VNC)

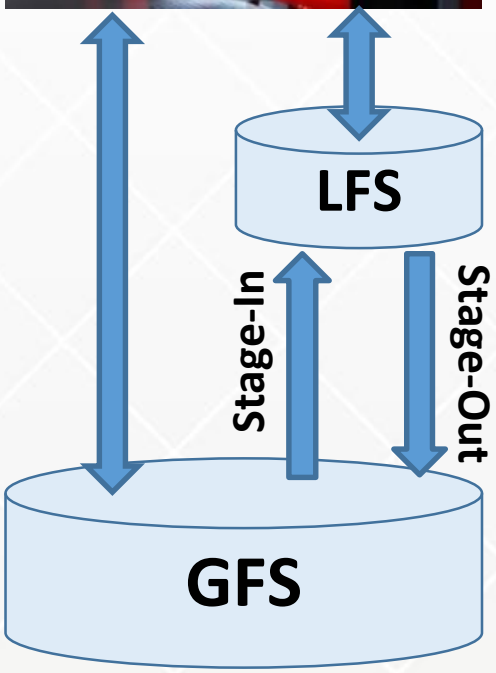
- **Batch Visualization (Up to 96 vCPUs)**



KVS (Kyoto Visualization System)

https://github.com/naohisas/KVS

K computer (GFS and LFS)

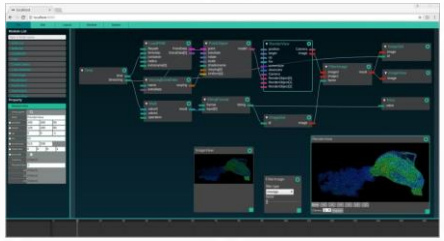


- **GFS**
  - Interactive mode
    - Up to 384 nodes
  - Batch mode (“micro” class job)
    - Up to 1152 nodes
- **LFS**
  - Batch mode
    - Up to 82,944 nodes (entire system)
  - Shared directory
  - Rank directory

HIVE (Multi-platform application)

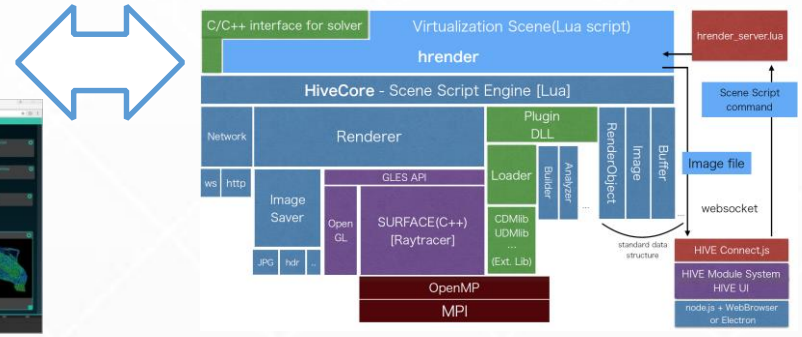
Computer simulations create the future

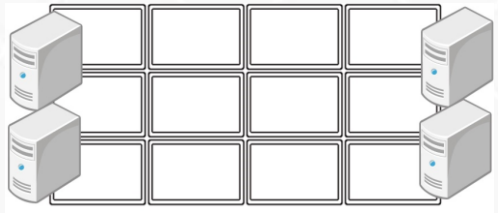
HIVE Client



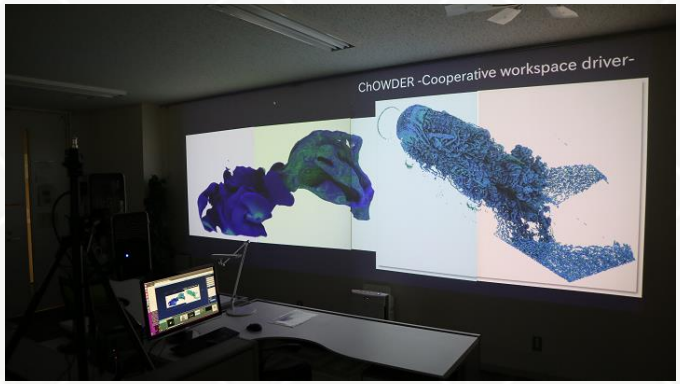
Web User Interface

Interactive Visual Exploration





Large Display System



HIVE Server

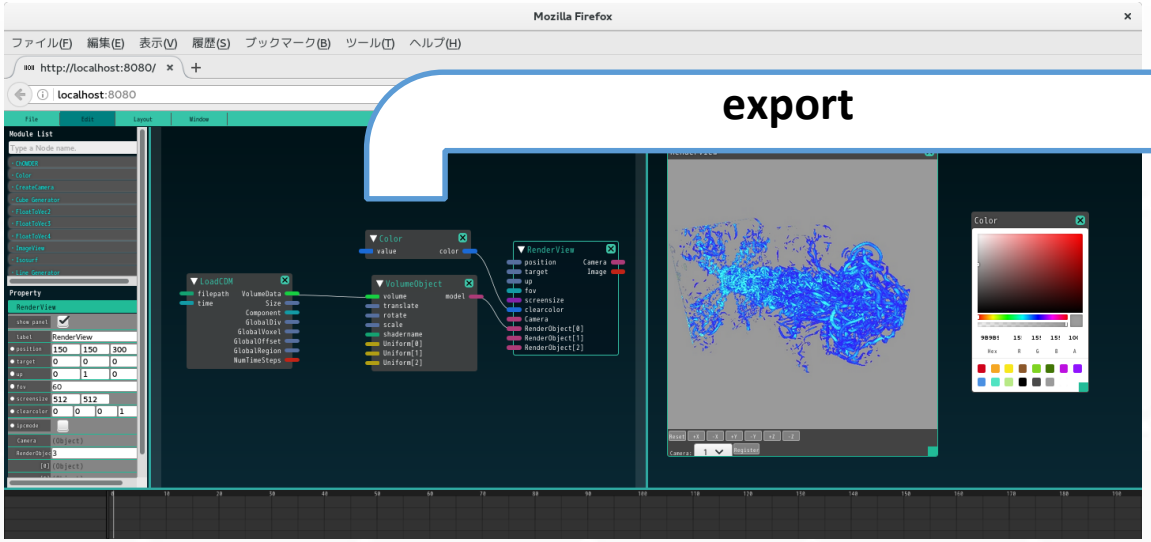
SPARC64
(K, FX10, FX100)

x86
(Server)
(Cluster)

Batch-based Large Data Visualization

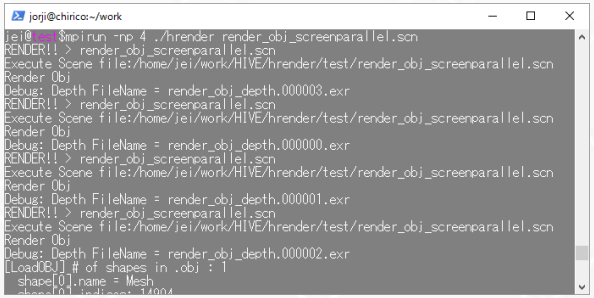
Batch-based large data visualization

Computer simulations create the future

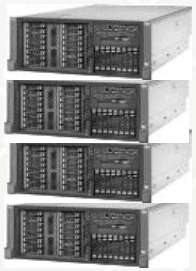



```
package.path = './?.lua;' .. package.path
local HiveBaseModule = require('HiveBaseModule')
local HIVE_ImageSaver = ImageSaver()
local HIVE_FRAMETIME=0
local HIVE_EXPORTMODE=true
local executedNode = {}
LoadCDM = {}
setmetatable(LoadCDM, {__index = HiveBaseModule})

LoadCDM.new = function (varname)
  local this = HiveBaseModule.new(varname);
  this.loader = require('CdmLoader')()
  setmetatable(this, {__index=LoadCDM})
  return this
end

function LoadCDM:Do()
  self:UpdateValue()
  if self.value.filepath ~= nil and self.value.filepath ~= "" then
    return self.loader:Load(self.value.filepath, self.value.time)
  else
    return false
  end
end
...
```

HIVE Server



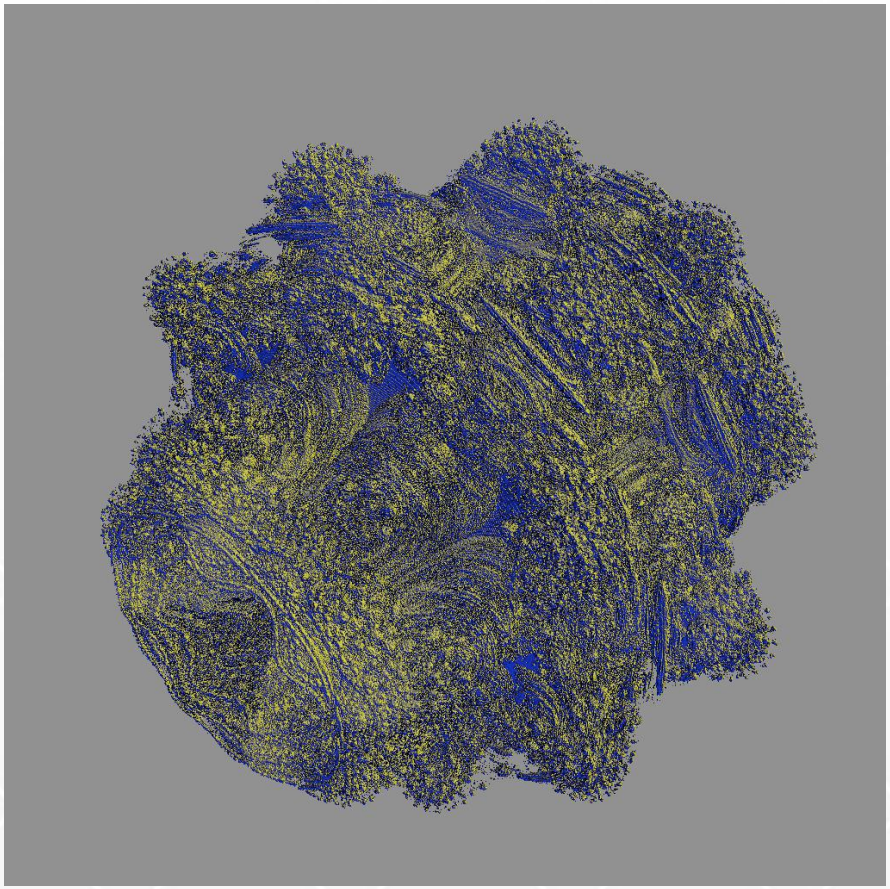
Command Line Interface

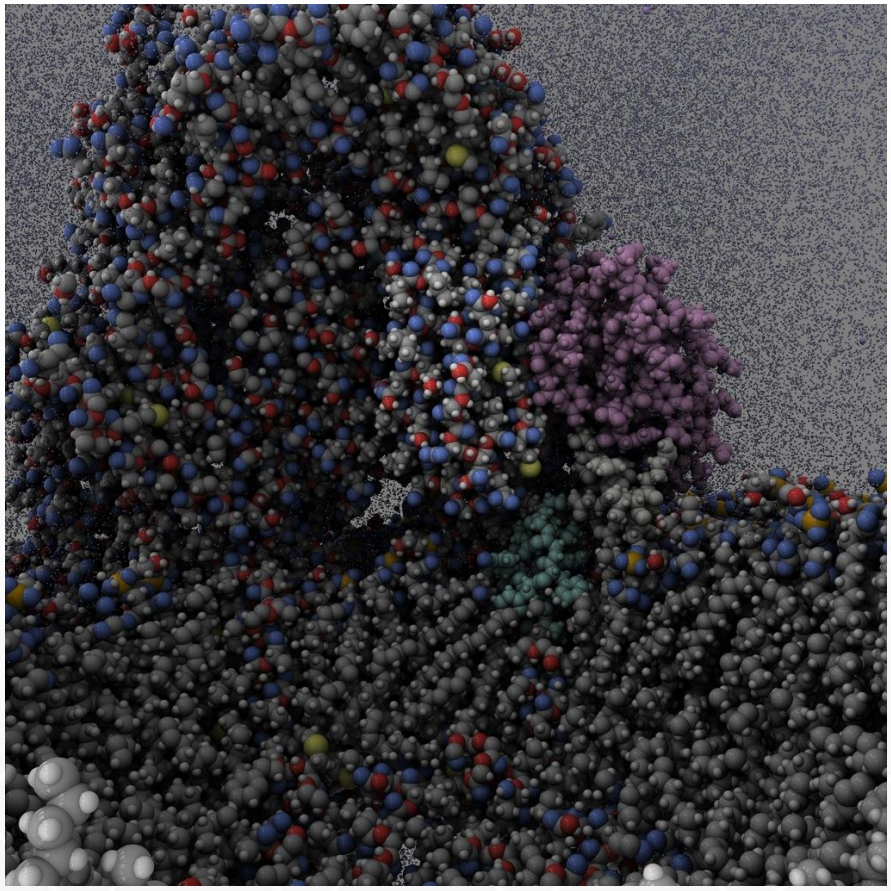


mpiexec -n X hrender Visual-Scene

HIVE/build/bin/hrender

Scalable rendering on the K computer





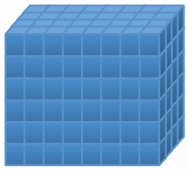
Ray Marching
3D Texture Size: 8192^3
Image Size: 16K8K
65,536 (2^{16}) nodes

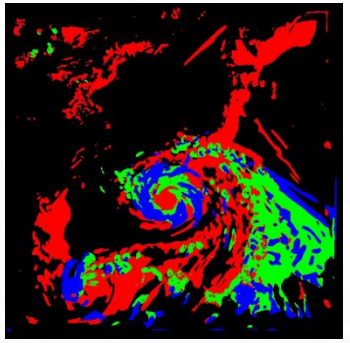
Ray Tracing
(288 x 288) [64x64 pixels]
Image Size: 18K18K (18,432)
82,944 nodes

Data Management Library (xDMlib)

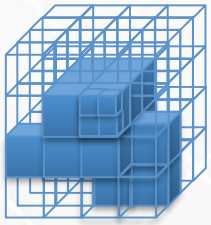
Computer simulations create the future

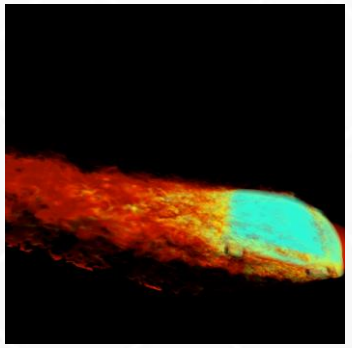
- **Cartesian Data (CDMlib)**
  - Voxel (SPH, VOL, RAW)
  - Structured (Plot3D XYZ, NetCDF SCALE)



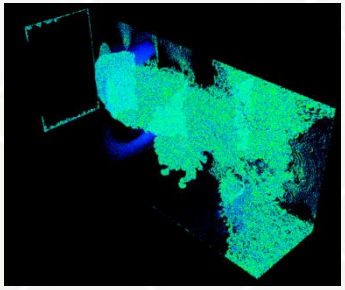


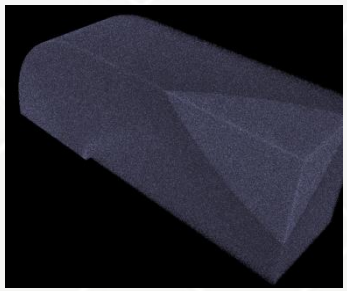
- **Hierarchical Multi-Resolution (HDMlib)**
  - Structured (BCM, ParaView PVTI)





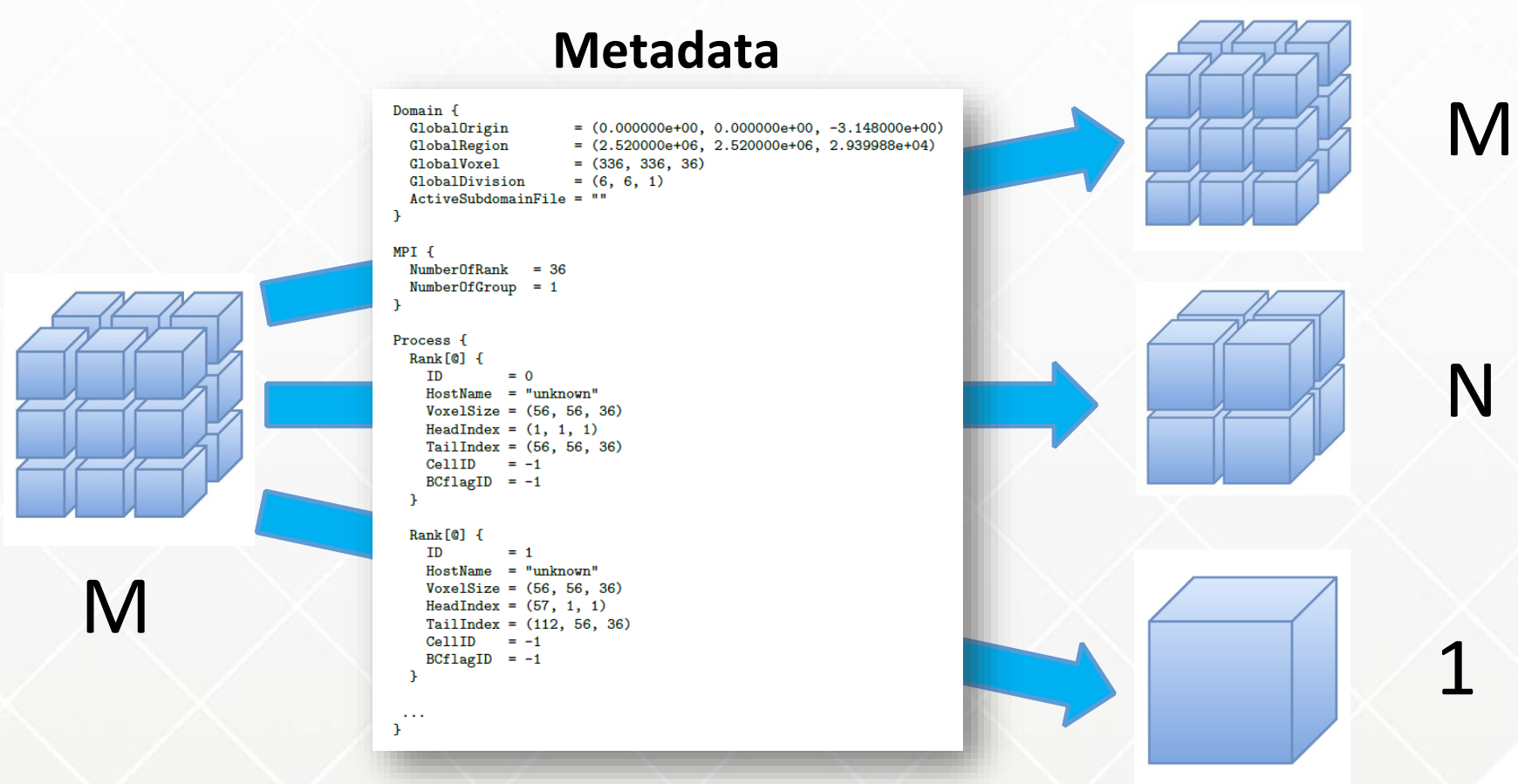
- **Unstructured (UDMlib)**
  - CFD General Notation (CGNS)
- **Particle (PDMlib)**





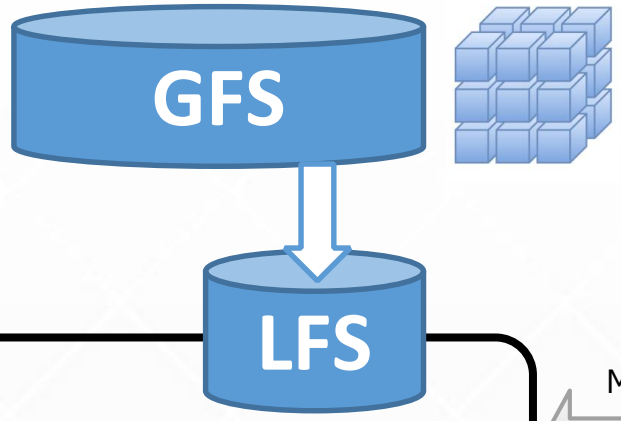
Flexible data loading mechanism

- **Cartesian data (CDMlib)**
  - Metadata (data/process mapping)



https://github.com/avr-aics-riken/CDMlib

Shared Directory



Computer simulations create the future

< Storage space at LFS >

/work

/JobName.JobID

Shared Directory

Meta-data
Shared dir.

MDS

Object-data
Shared dir.

OSS

Processes

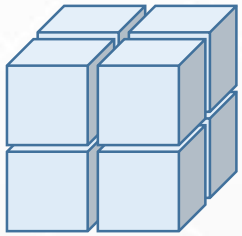
Rank #0 | Rank #1 | Rank #2 | ..... | Rank #(N-1)

Accesses to a shared directory

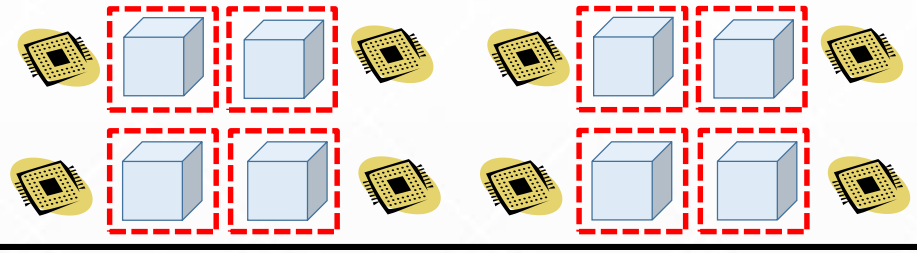
Shared directory (CDMlib data loading)

Computer simulations create the future

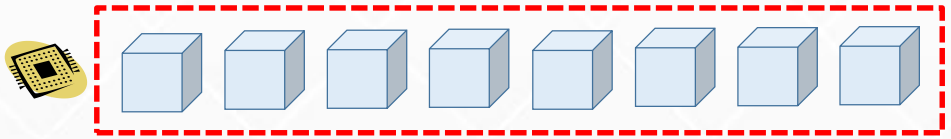
M = 8
(2 x 2 x 2)



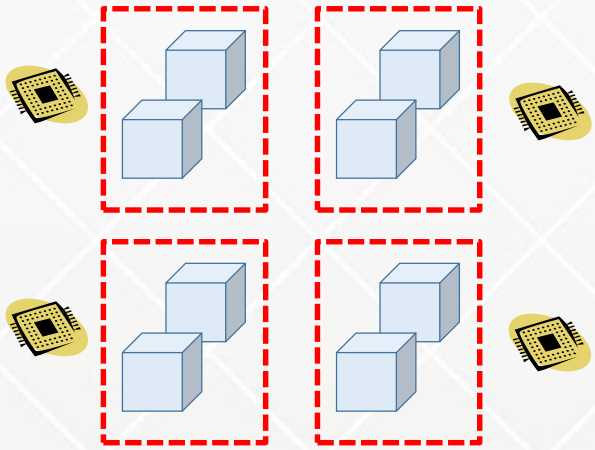
M x M
N = 8
(2 x 2 x 2)



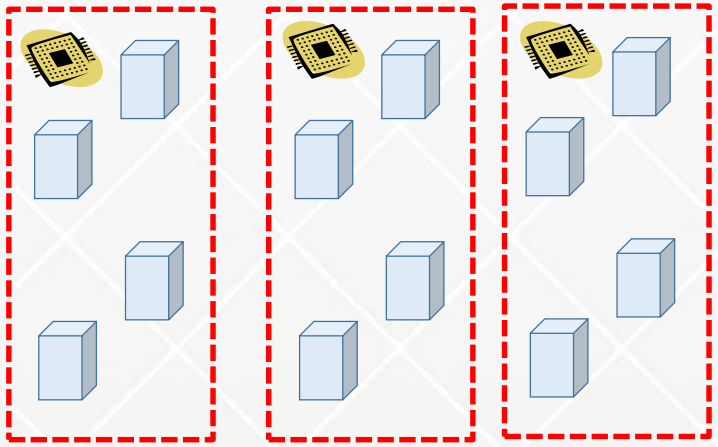
M x 1
N = 1
(1 x 1 x 1)



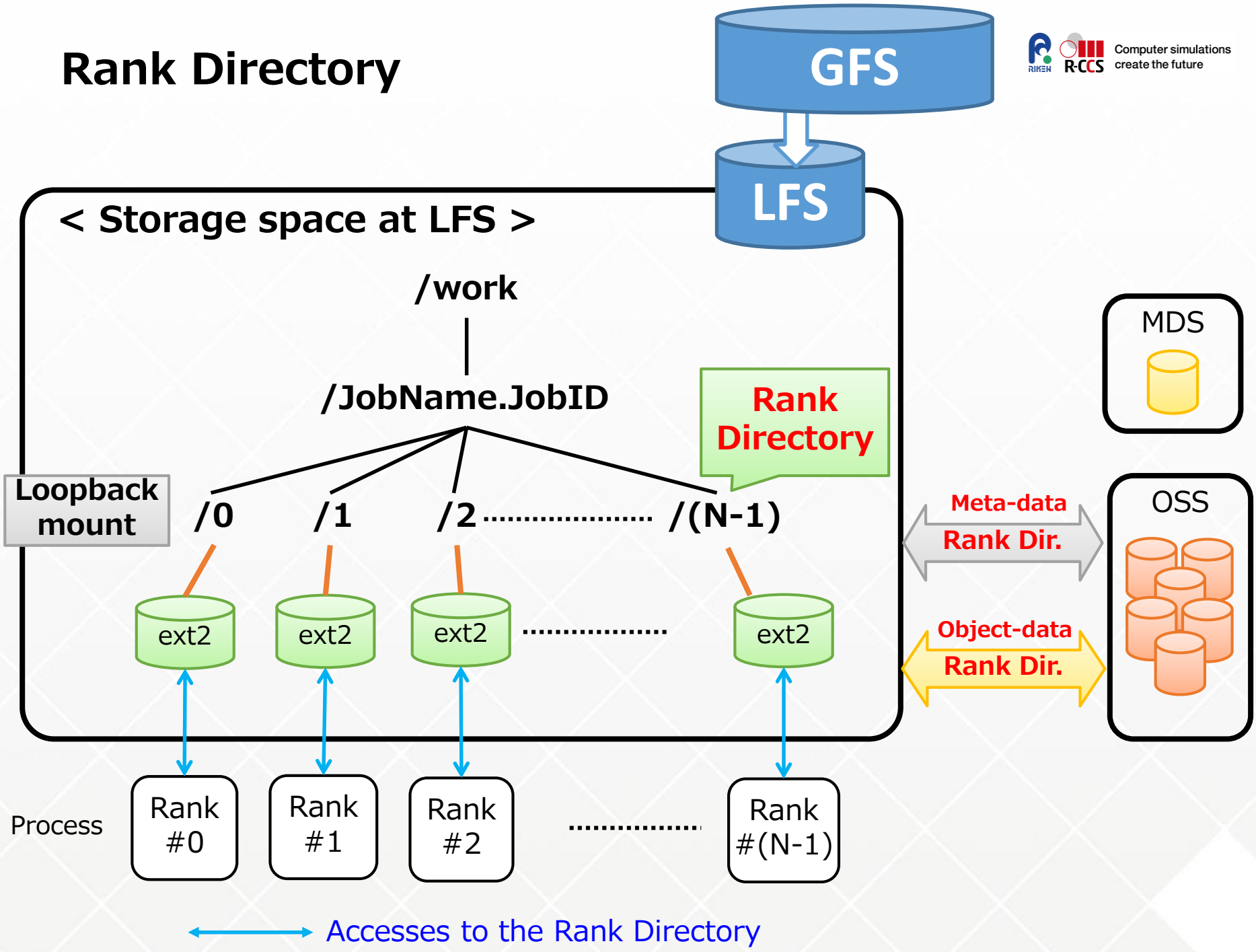
M x N [Aligned]
N = 4 (4 x 1 x 1)



M x N [Non-Aligned]
N = 3 (3 x 1 x 1)

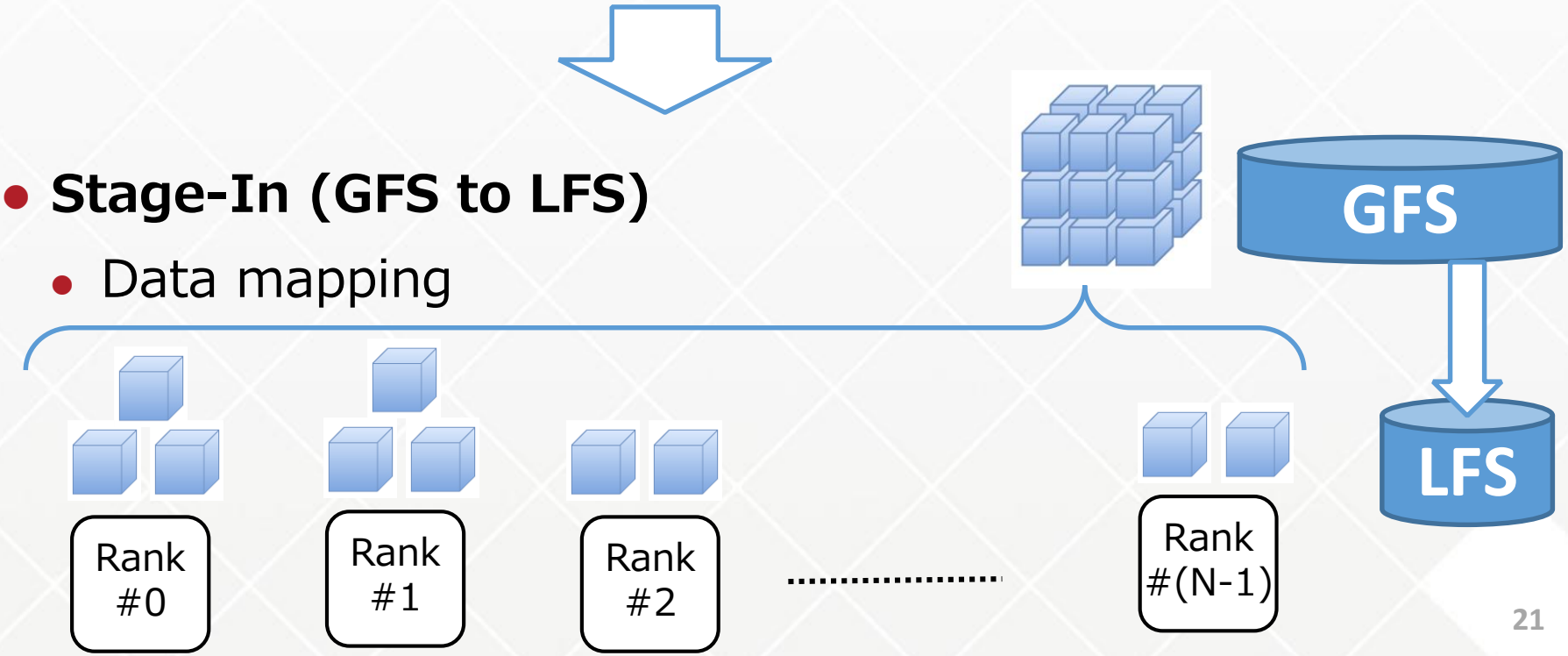


Rank Directory



Rank Directory (Loopback File System)

- **Mitigation of I/O interference among processes**
- **Reducing the MDS accesses leads to effective utilization of LFS.**
- **I/O accesses in Rank Directories are free from slowdown of MDS performance.**

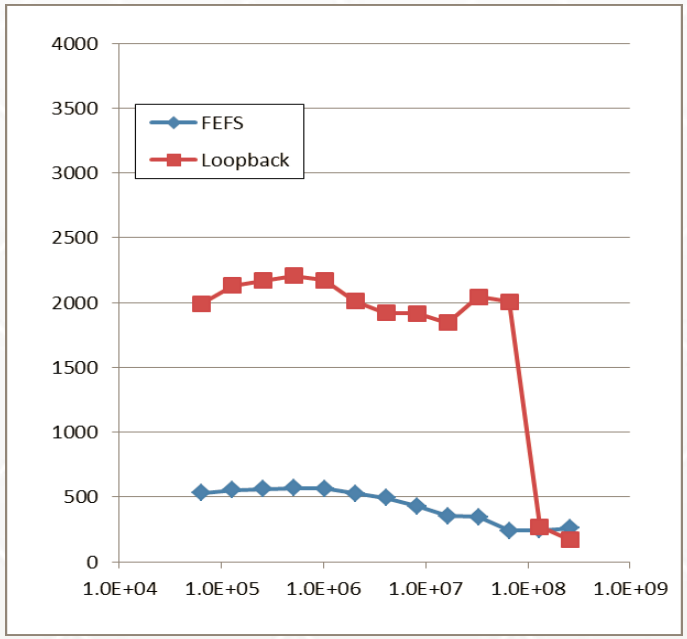


- **Stage-In (GFS to LFS)**
  - Data mapping

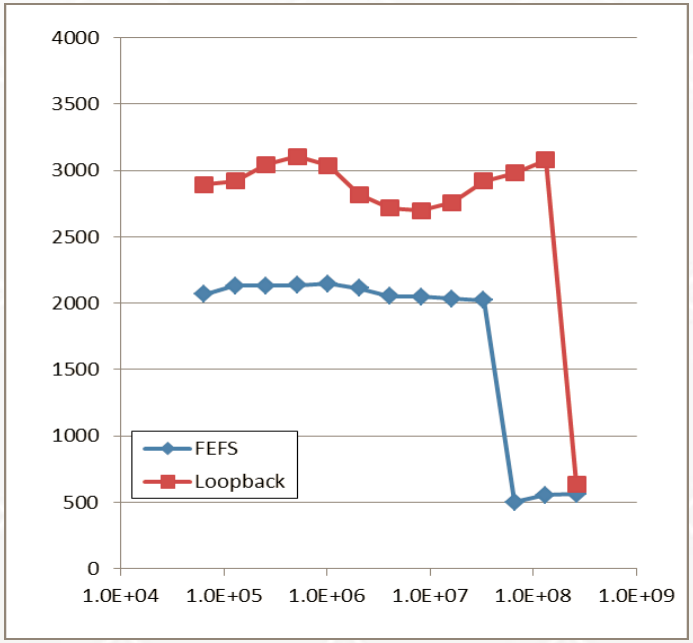
I/O performance evaluation

- **IOzone (Single node I/O performance evaluation)**
  - FEFS (Shared directory among nodes) vs Loopback
  - Loopback (Rank directory) outperformed FEFS for smaller data size with the help of file system cache

write (64KB I/O block)



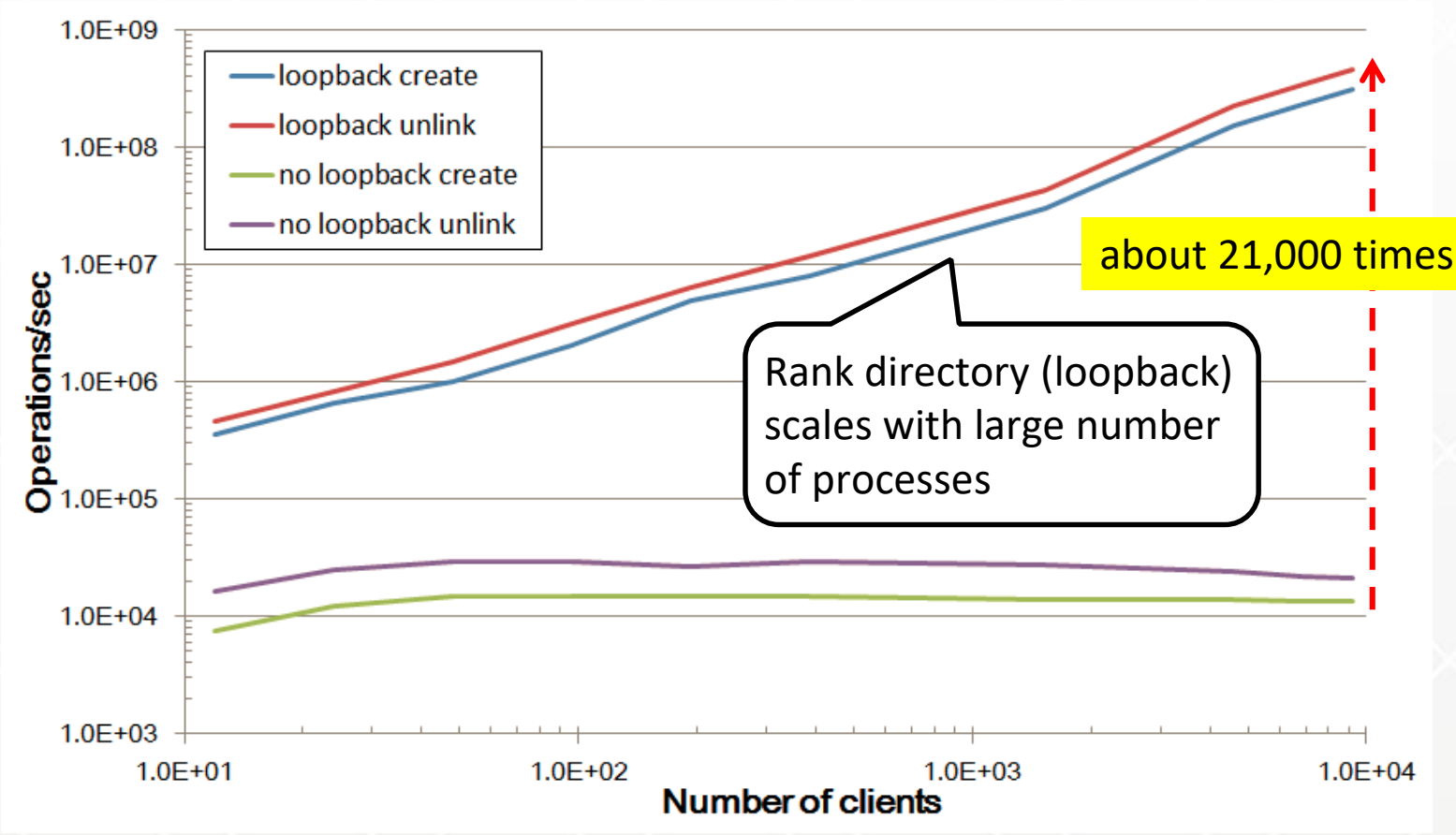
read (64KB I/O block)



K. Yamamoto F. Shoji, A. Uno, S. Matsui, K. Sakai, F. Sueyasu, and S. Sumimoto,
"Analysis and Elimination of Client Evictions on a Large Scale Lustre Based File System," LUG'15

Total metadata access performance

- **mdtest (100 files/node)**
  - Create 26K ops/node; Unlink 37K ops/node

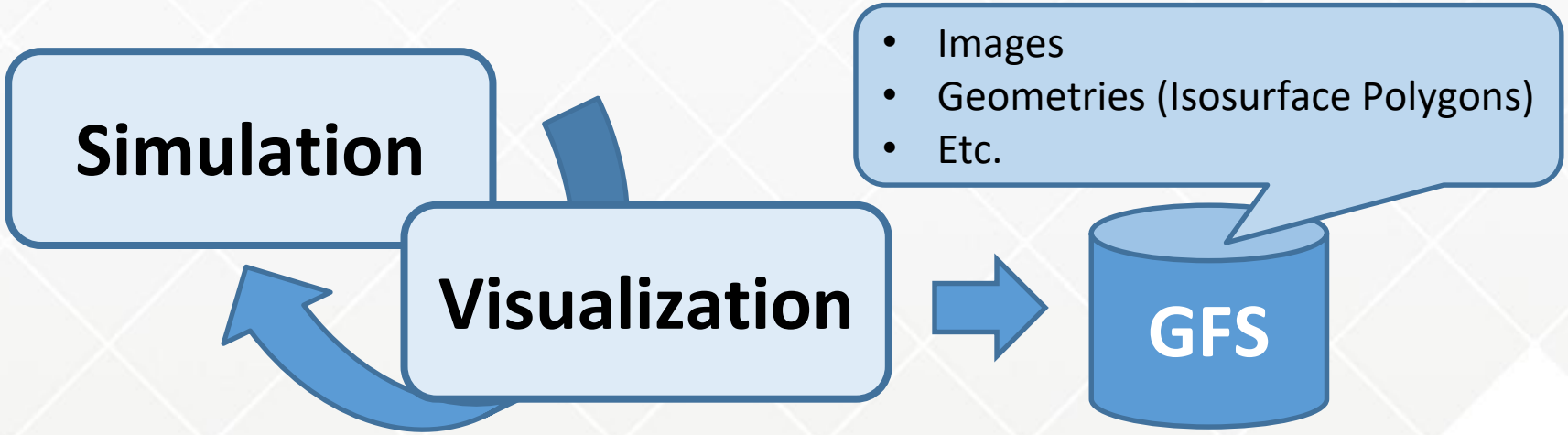


K. Yamamoto, F. Shoji, A. Uno, S. Matsui, K. Sakai, F. Sueyasu, and S. Sumimoto,
"Analysis and Elimination of Client Evictions on a Large Scale Lustre Based File System," LUG'15

Large data visualization

- **Post-hoc Visualization**

- **In-situ Visualization**



In-Situ Visualization

- **OpenFOAM (CFD Simulation) + KVS Library (PBVR)**

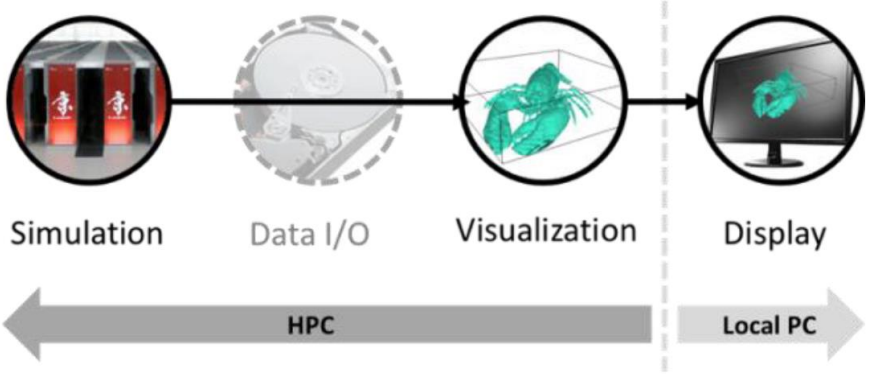
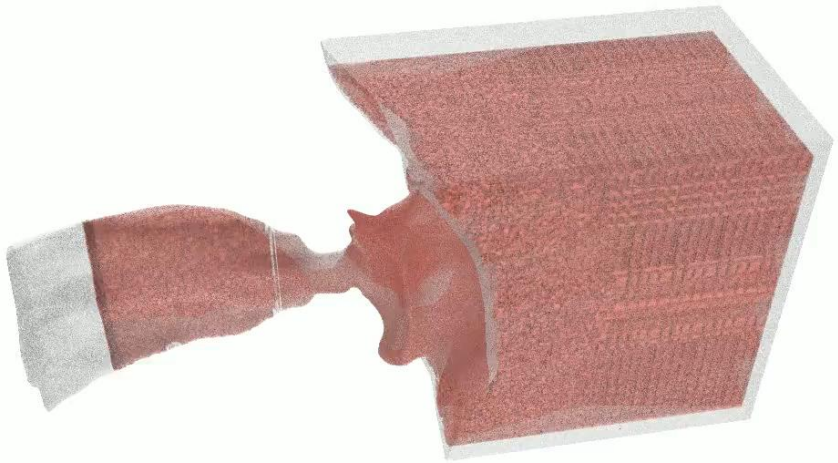
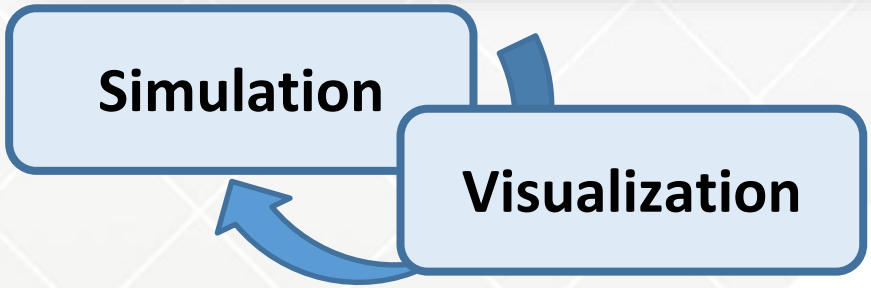


Fig. 1 In-situ visualization processing pipeline.

```
1.  //simulation code
2.     float data_array[];
3.     float coords[]; // Mesh coordinate information
4.     int connection[]; //  Mesh connection information
5.
6.     init_simulation();
7.
8.     while( now_time == end_time )
9.     {
10.      solver( data_array, coords, connection );
11.      insitu_vis( data_array, data_size, coords,
    connection, now_time );
12.    }
13.
14.    end_simulation();
```

Fig. 2 Sample simulation code inserting in-situ visualization function.





Same Process

Acknowledgements

- **Operations and Computer Technologies Division**
  - Fumiyoshi Shoji (Division Director)
  - Atsuya Uno (Unit Leader)
  - Masaaki Terai (R&D Researcher)
- **Kyushu University**
  - Kenji Ono
- **Kobe University**
  - Naohisa Sakamoto
- **University of Utah**
  - Valerio Pascucci
  - Steve Petruzza