

HIVE

講習会

Jorji NONAKA, Tomohiro KAWANABE
RIKEN AICS

2016年10月5日

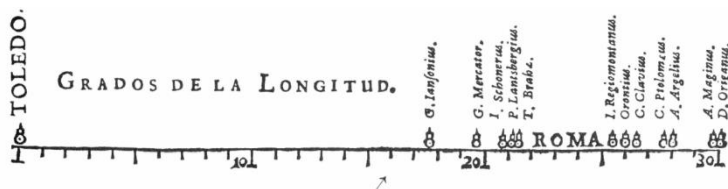
Heterogeneously Integrated Visual analytics Environment

From: Wolfram Alpha



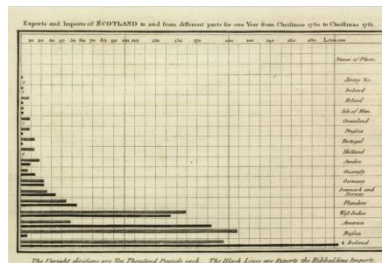
Cave Painting
15,000 BC

From: Milestones in the History of Data Visualization

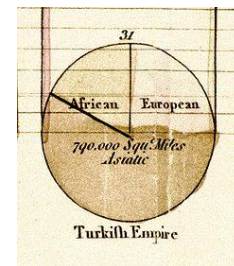


Data Visualization
1644

From: Wikipedia (William Playfair)



Bar Chart
1786



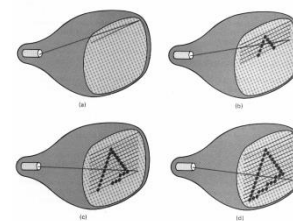
Pie Chart
1801



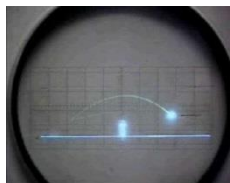
Whirlwind 1951



Sketchpad 1963



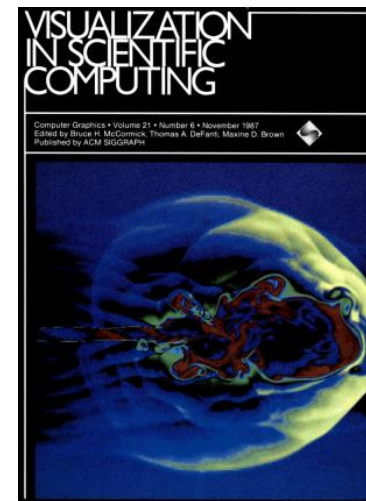
Raster Graphics
1970



Tennis for Two
1958

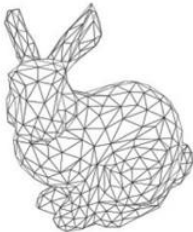


ACM's Special Interest Group
in Computer Graphics and
Interactive Techniques
1969



Scientific Visualization
1987

Visual Analytics



Stanford University
Spring 2003 Rendering Competition



ACM SIGGRAPH
Special Interest Group on
Computer Graphics and
Interactive Techniques

Computer Graphics



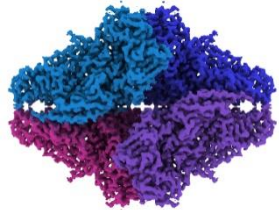
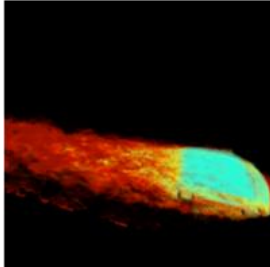

IEEE
Advancing Technology
for Humanity

Workshop on Scientific Visualization

1989

Scientific Visualization

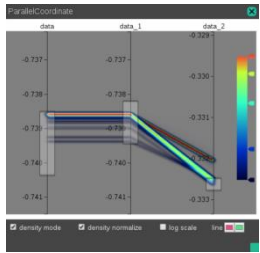
IEEE SciVis



1995

Information Visualization

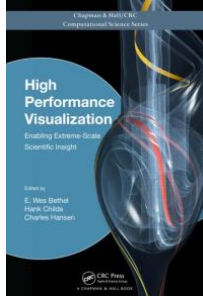
IEEE InfoVis



2006

Visual Analytics

IEEE VAST



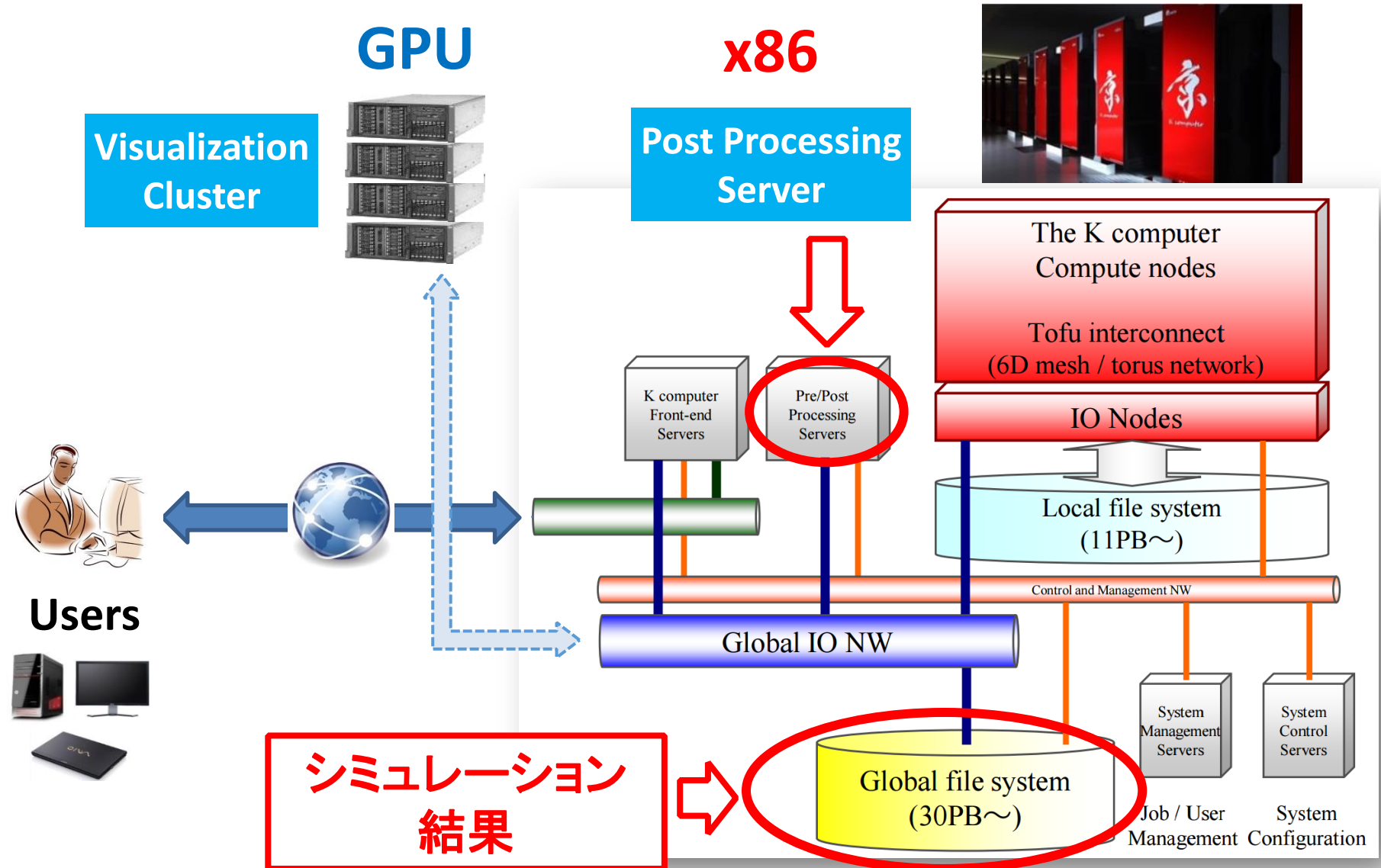
HIVE

High Performance Visualization

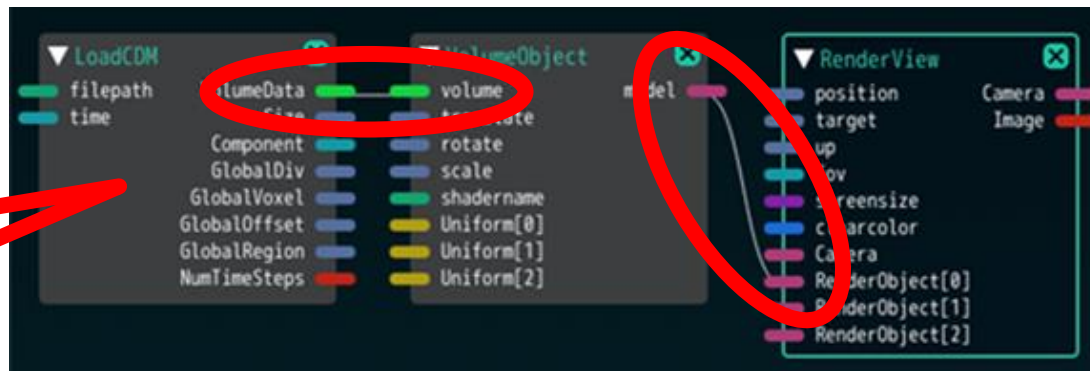
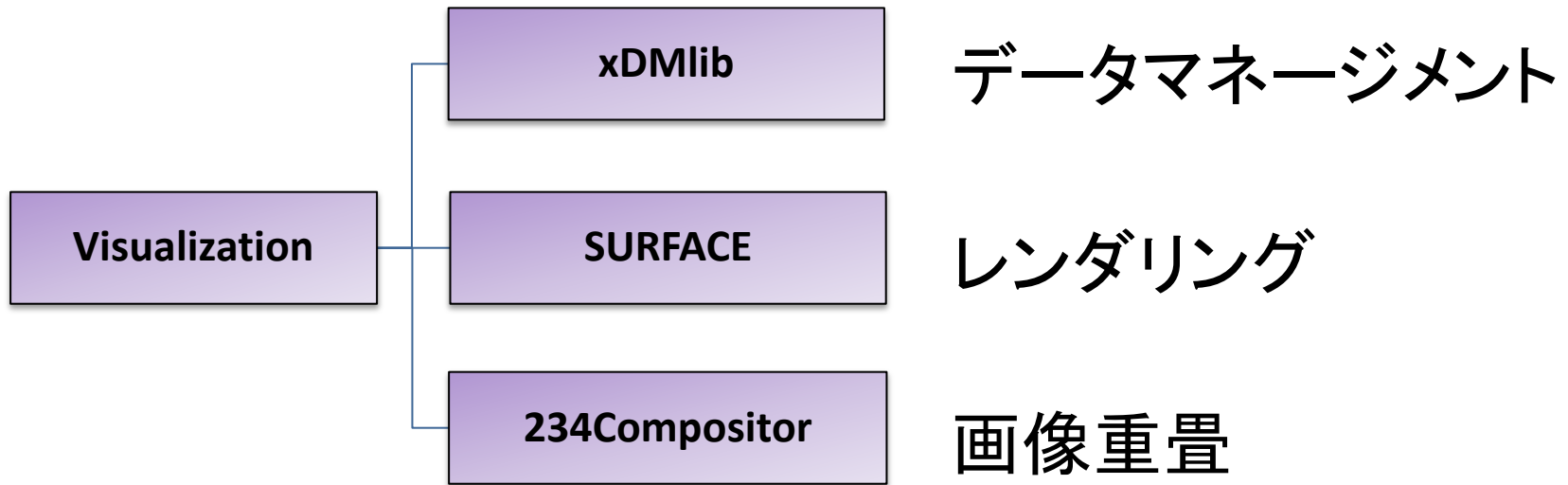


Heterogeneous HPC Infrastructure

SPARC64



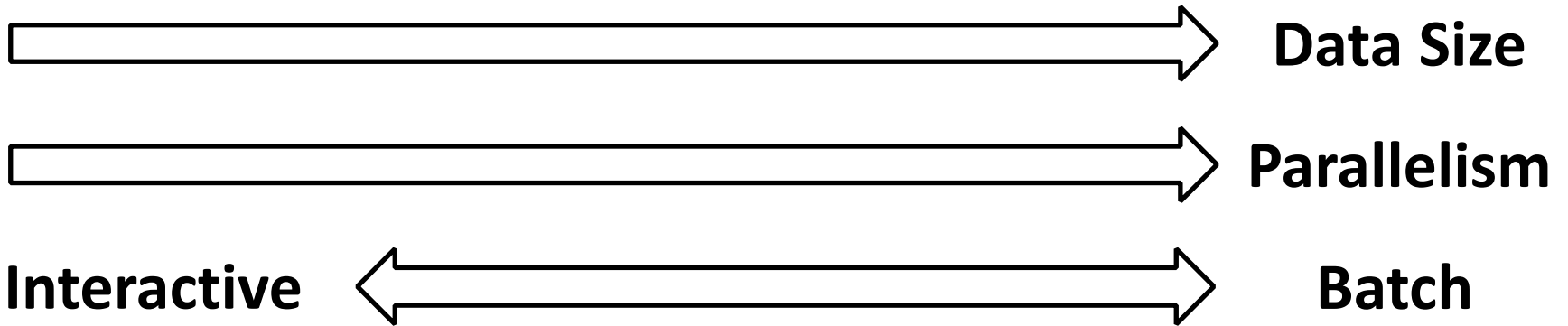
可視化ワークフロー



ノード

処理の流れ

Integration



- サーバークライアント型

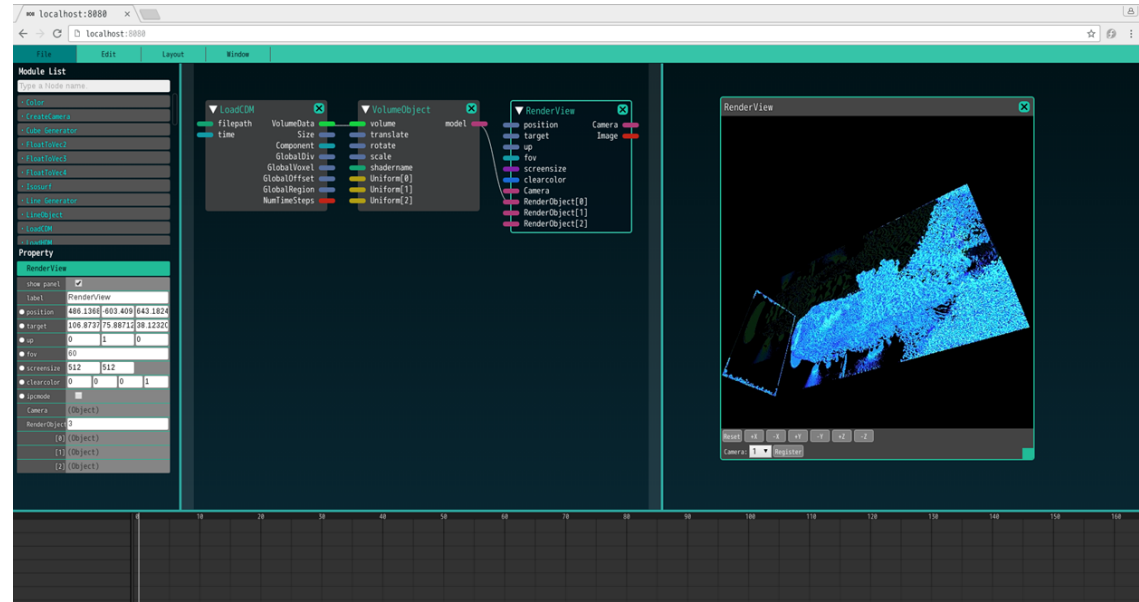


Stand-Alone (Local Server) Mode



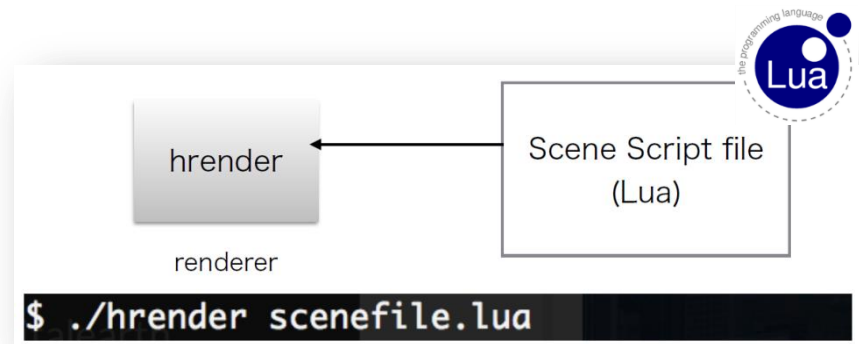
HIVE Client

HIVE Server



Web-based User Interface

Linux
-CentOS
-Ubuntu
Mac OSX



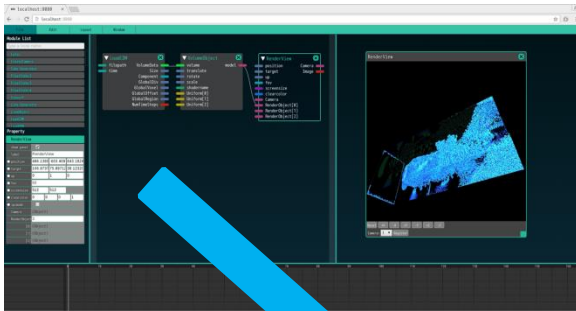
Command Line Interface

Remote Server Mode



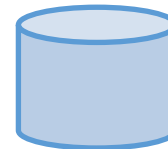
HIVE Client

HIVE Server



`mpixec -n X HIVE-Render Visual-Scene`

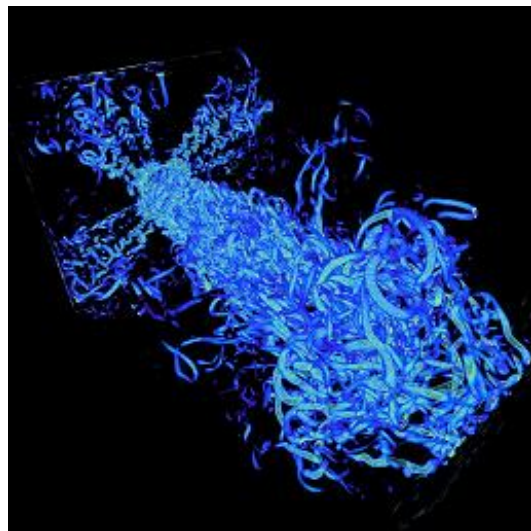
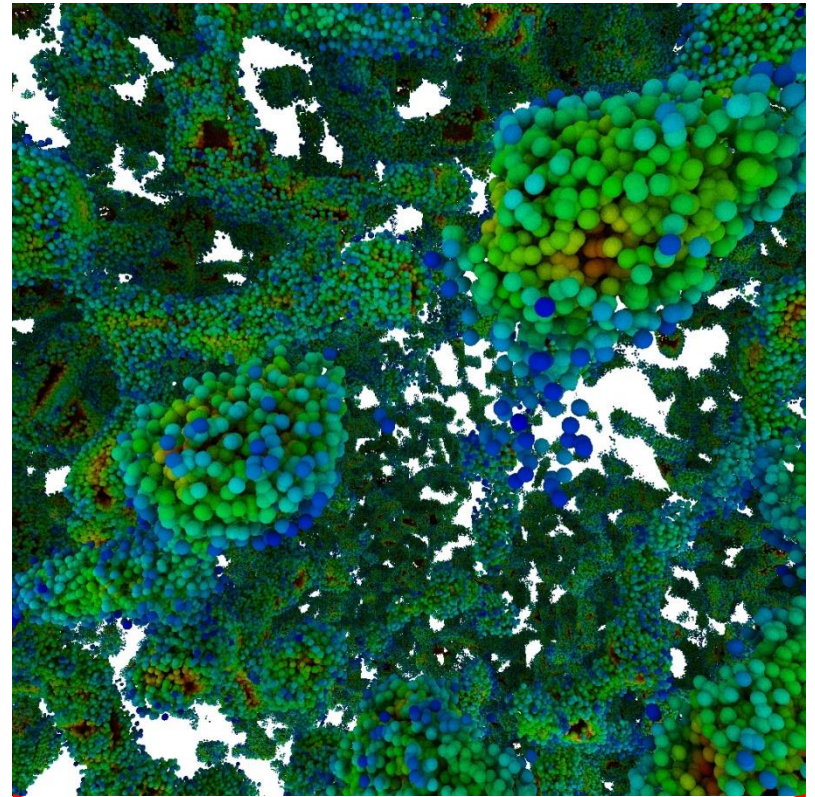
```
local instSPHLoader1 = LoadSPH({filepath='/work/HIVE_DATA/SPH_DATA/OUT/out.sph'})
local instVolumeModel2 = CreateVolumeModel({volume=instSPHLoader1:VolumeData(),
translate={0,0,0}, rotate={0,0,0}, scale={1,1,1},
shadername='/work/HIVE_DATA/SPH_DATA/OUT/volume_out.sph.frag', Uniform={nil, nil,
nil}})
local instCreateCamera3 = CreateCamera({position={100,-300,300}, target={0,0,0}, up={0,1,0},
fov = 60, screensize={512,512}, clearcolor={0,0,0,1}, color_file='output.jpg', nil})
local renderConsole4 =
RenderToConsole({RenderObject={RenderObject0=instCreateCamera3:camera(),
RenderObject1=instVolumeModel2:model(), nil}})
```



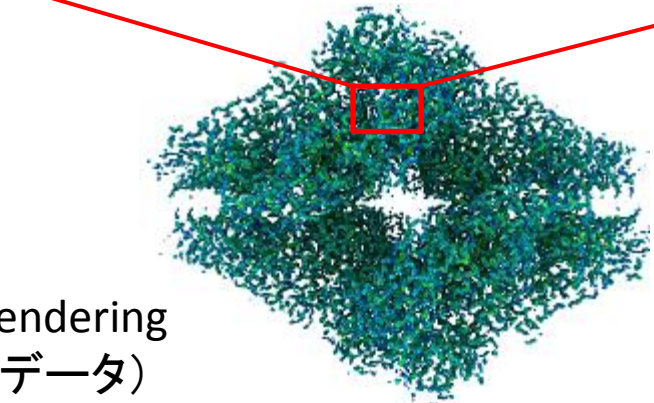
SURFACE (レンダリング)



Volume Rendering (半透明データ)



Surface Rendering
(不透明データ)



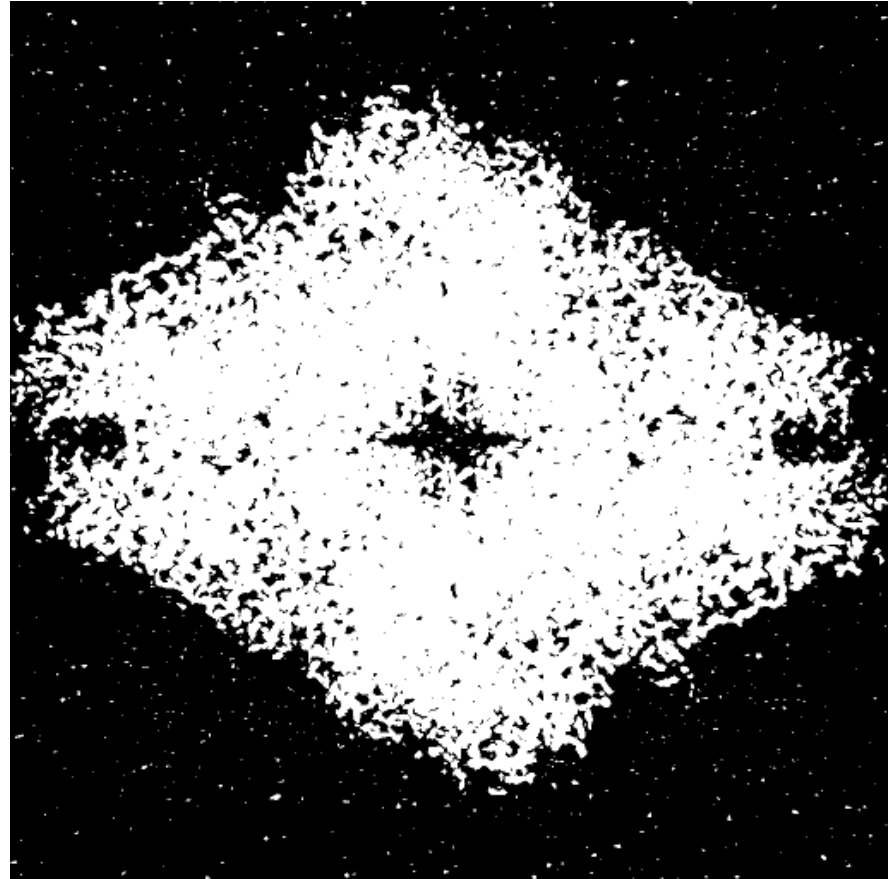
GLSL Shader

```
#ifdef GL_ES
precision mediump float;
#endif

uniform sampler2D tex0;
uniform vec2 resolution;
varying float matID;
varying vec3 mnormal;
uniform vec4 color;

void main(void)
{
    gl_FragColor = vec4(1,1,1,1);
    return;
}
```

white.frag



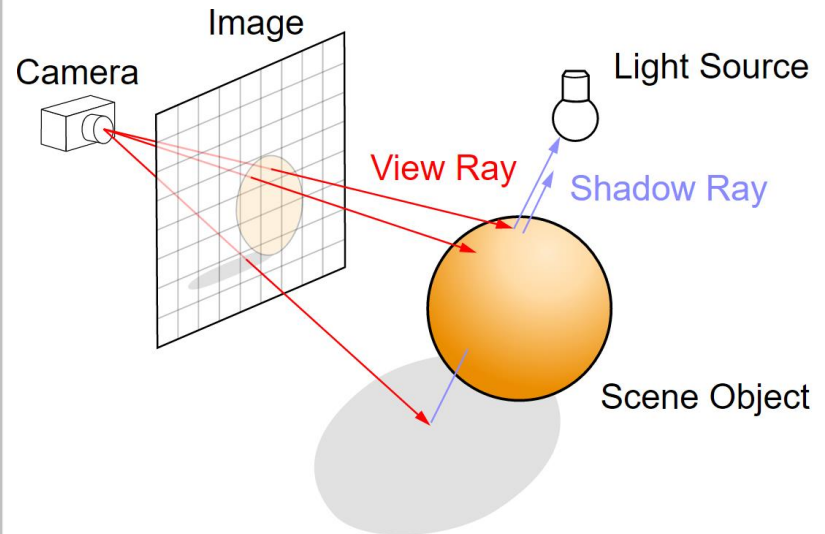
GLSL Shader (Trace Extensions)

```
#extension GL_LSGL_trace

// Raytracing function
float trace(vec3 pos, vec3 dir, out vec4 Cs)

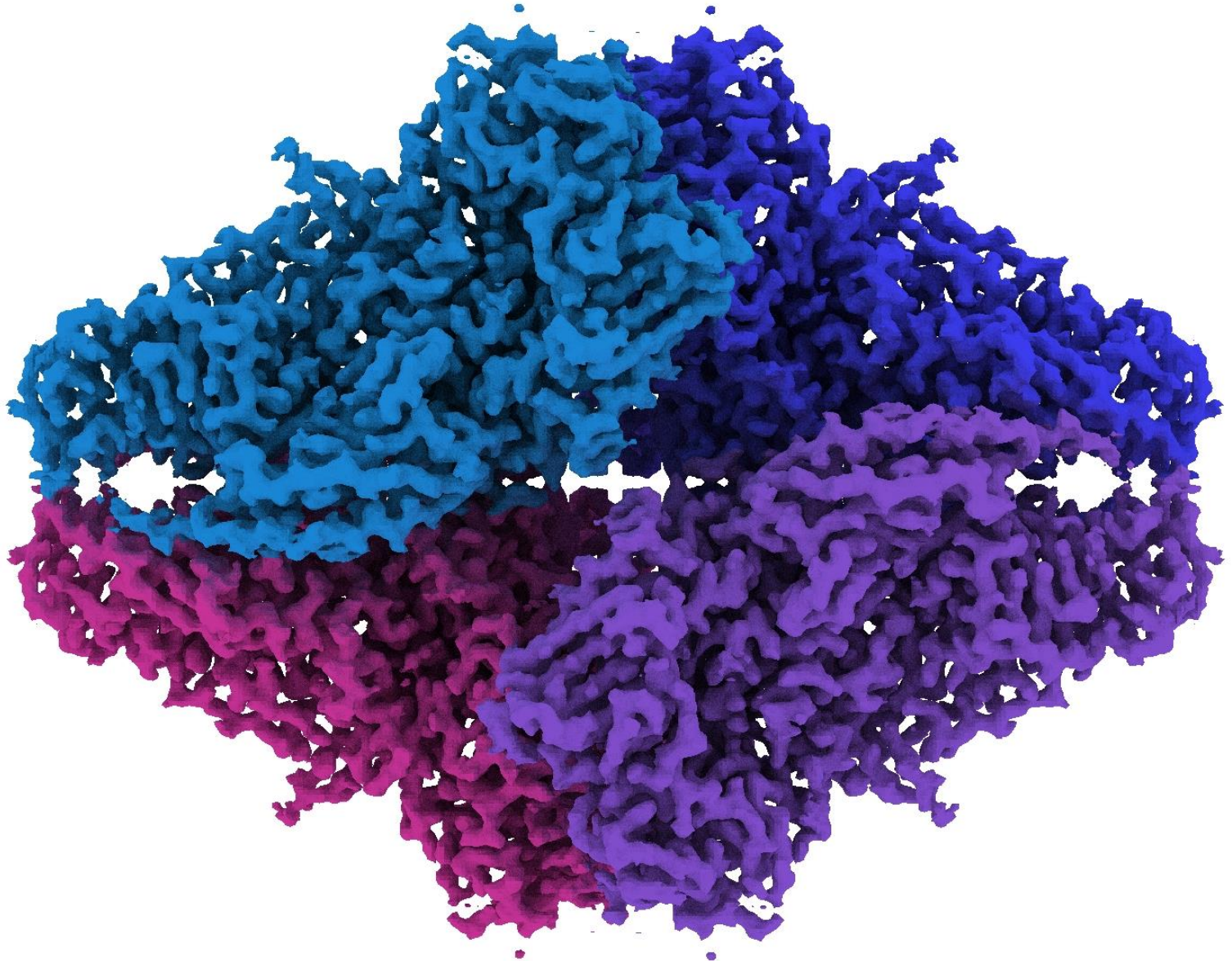
// Get intersection information: distance,
// position, normal and incident ray direction
float isectinfo (out vec3 p, out vec3 n, out vec3
indir);

// Returns # of reflection events for the ray.
int raydepth (out int depth);
```



[https://en.wikipedia.org/wiki/Ray_tracing_\(graphics\)](https://en.wikipedia.org/wiki/Ray_tracing_(graphics))

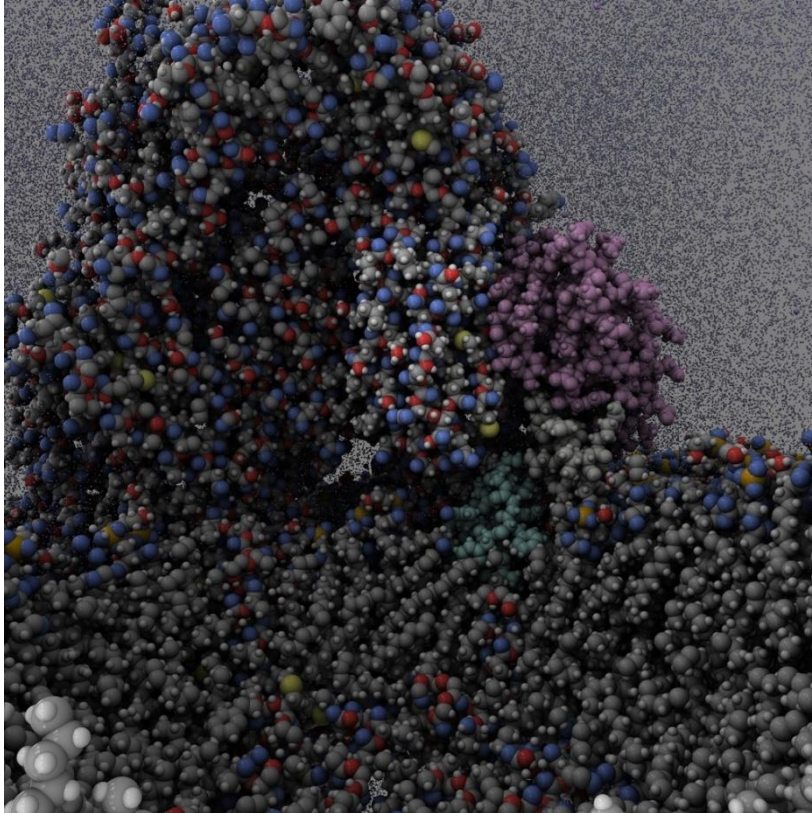
GLSL シェーダー



SURFACE（並列レンダリング）

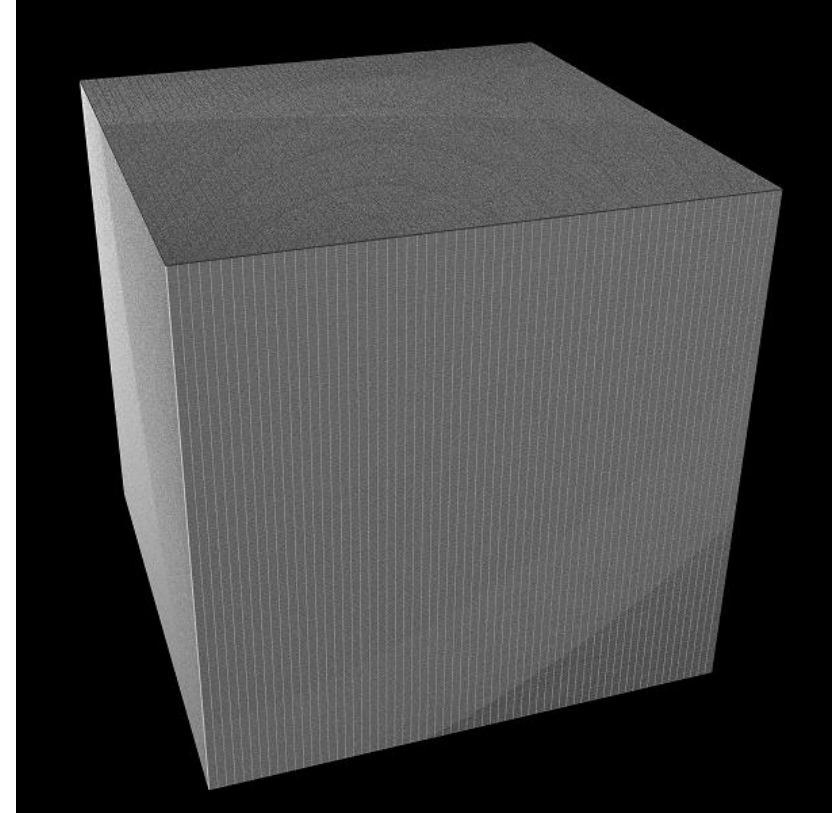


(2D) 画像空間分割



82,944 ノード
16K x 8K イメージ

(3D) データ空間分割



65,536 (2^{16}) ノード
8K x 8K イメージ

HIVEモジュールシステム



HIVE
Server

サーバー:ポート番号

The screenshot shows a web browser window with the URL 'chirico:8080'. The interface is divided into several sections:

- Node View:** A list of modules on the left, including ChOWDER, Color, CreateCamera, Cube Generator, FloatToVec2, FloatToVec3, FloatToVec4, ImageView, Isosurf, Line Generator, and LineObject. Below this is a 'Property' section for the selected 'Color' module, showing a 'value' property with a grid of 1s.
- Panel View:** A 'Color' panel on the right with a color picker, a spectrum, and a hex input field showing '#FFFFFF'.

プロパティ

Node View

Panel View

モジュール

ノード

パネル

モジュール
検索

操作

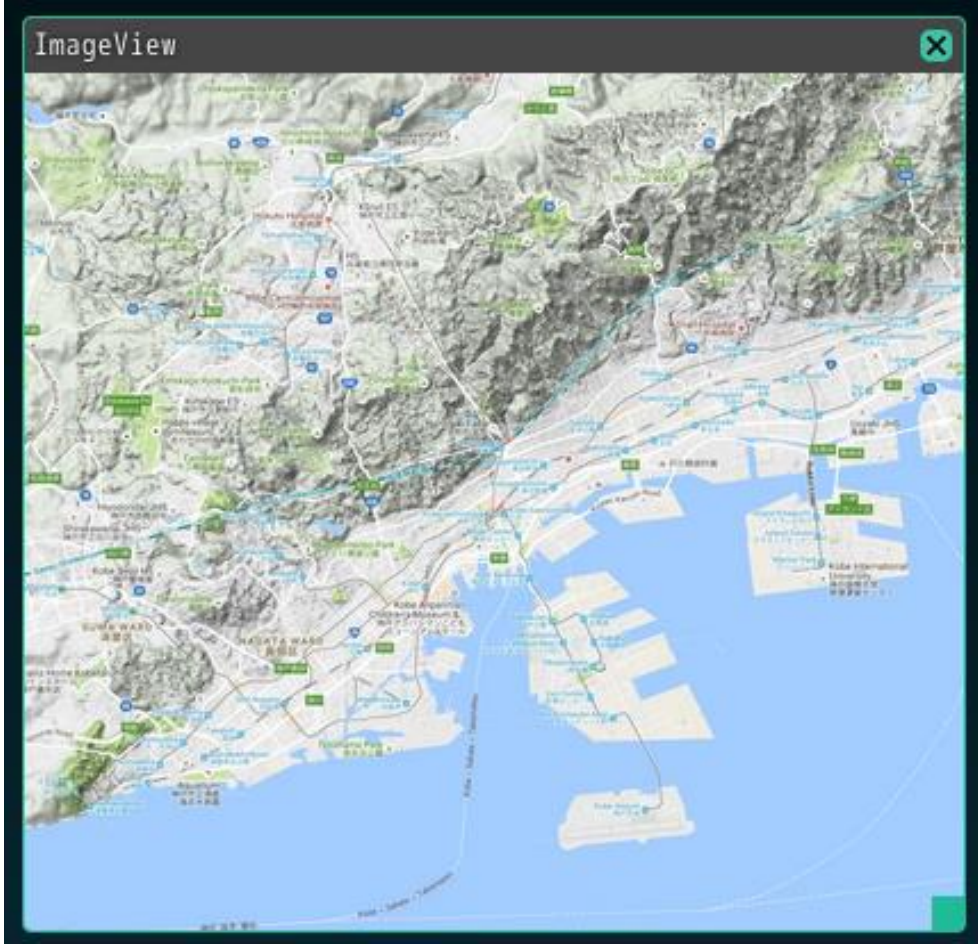
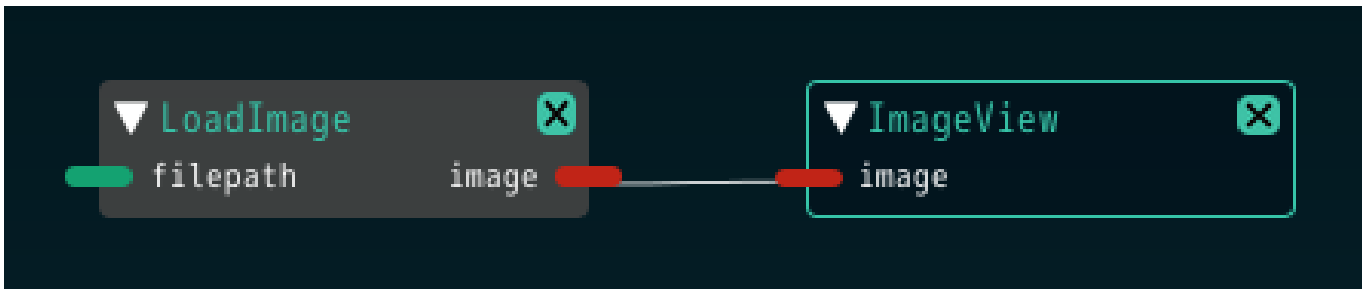
The screenshot shows a software interface with a dark theme. On the left, there is a 'Module List' panel with a search bar and a list of modules including ChOWDER, Color, CreateCamera, Cube Generator, FloatToVec2, FloatToVec3, FloatToVec4, ImageView, Isosurf, Line Generator, and LineObject. Below this is a 'Property' panel for the 'Color' module, showing a 'value' property with four input fields, each containing the number '1'. A callout bubble points to the search bar in the module list. In the center, a 'Color' property is shown with a dropdown arrow, the text 'Color', and a 'value' field. A callout bubble points to this property. On the right, a 'Color' dialog box is open, showing a color selection interface with a gradient bar, a rainbow spectrum, and a hex code input field containing '#FFFFFF'. A callout bubble points to the right side of the dialog. At the bottom, a grid is visible with numerical labels from 40 to 110.

ビュー

平行移動
右ボタン+左ボタン

モジュールリスト
- ダブルクリック
- スペースキー

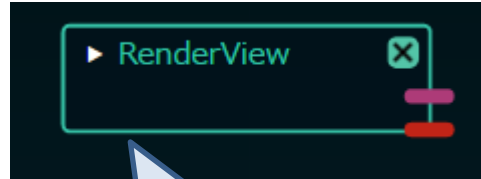
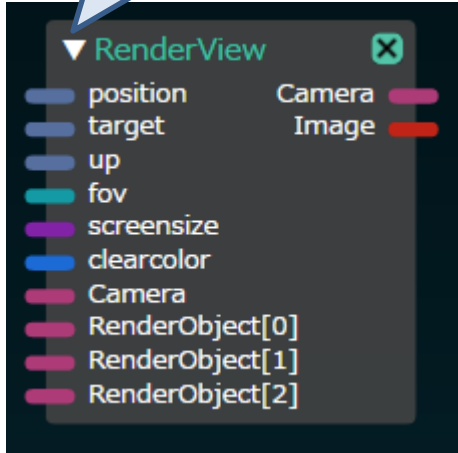
拡大・縮小
右ボタン



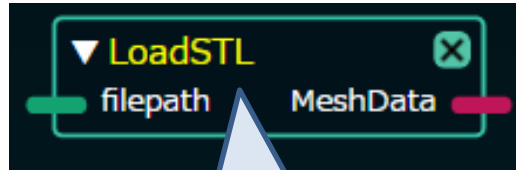
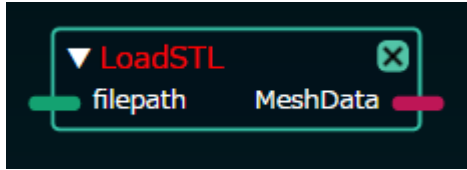
/home1/giud/share/HIVE/seminar_data/Sample/kobe.jpg

操作

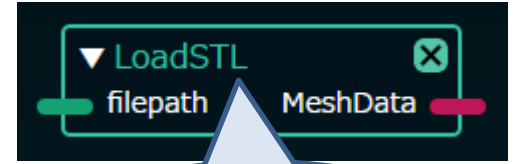
簡略化
表示



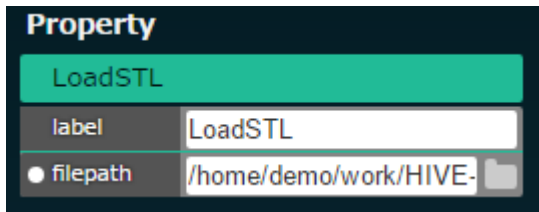
選択状態
のノード



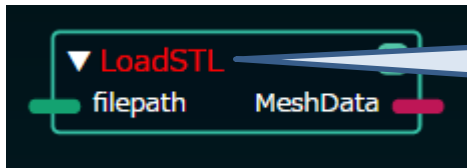
実行中



正常に
終了



実行に
失敗

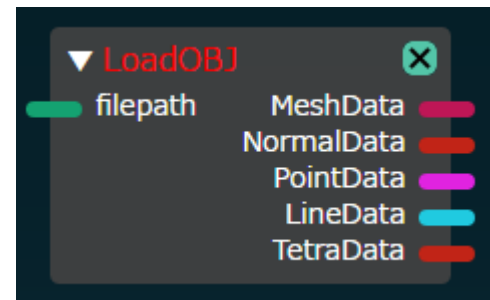
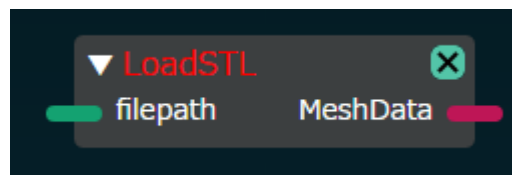


データ読み込み

● Loadモジュール

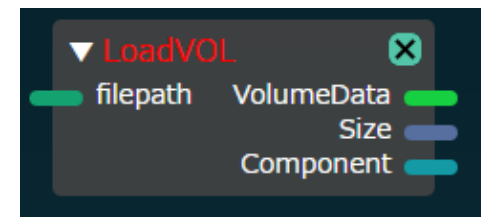
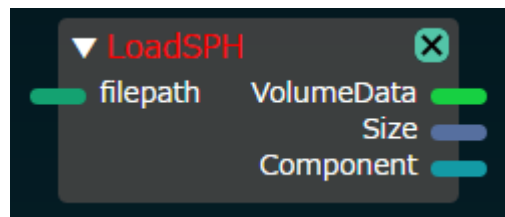
● メッシュデータ

- STLフォーマット
- OBJフォーマット

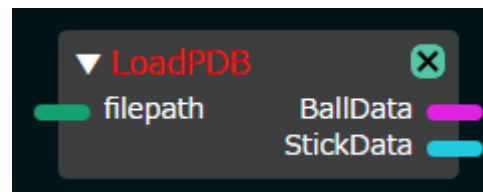


● ボリュームデータ

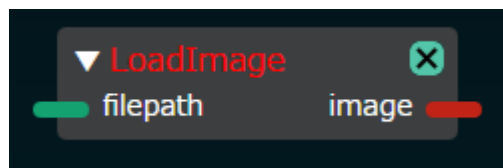
- SPHフォーマット
- VOLフォーマット

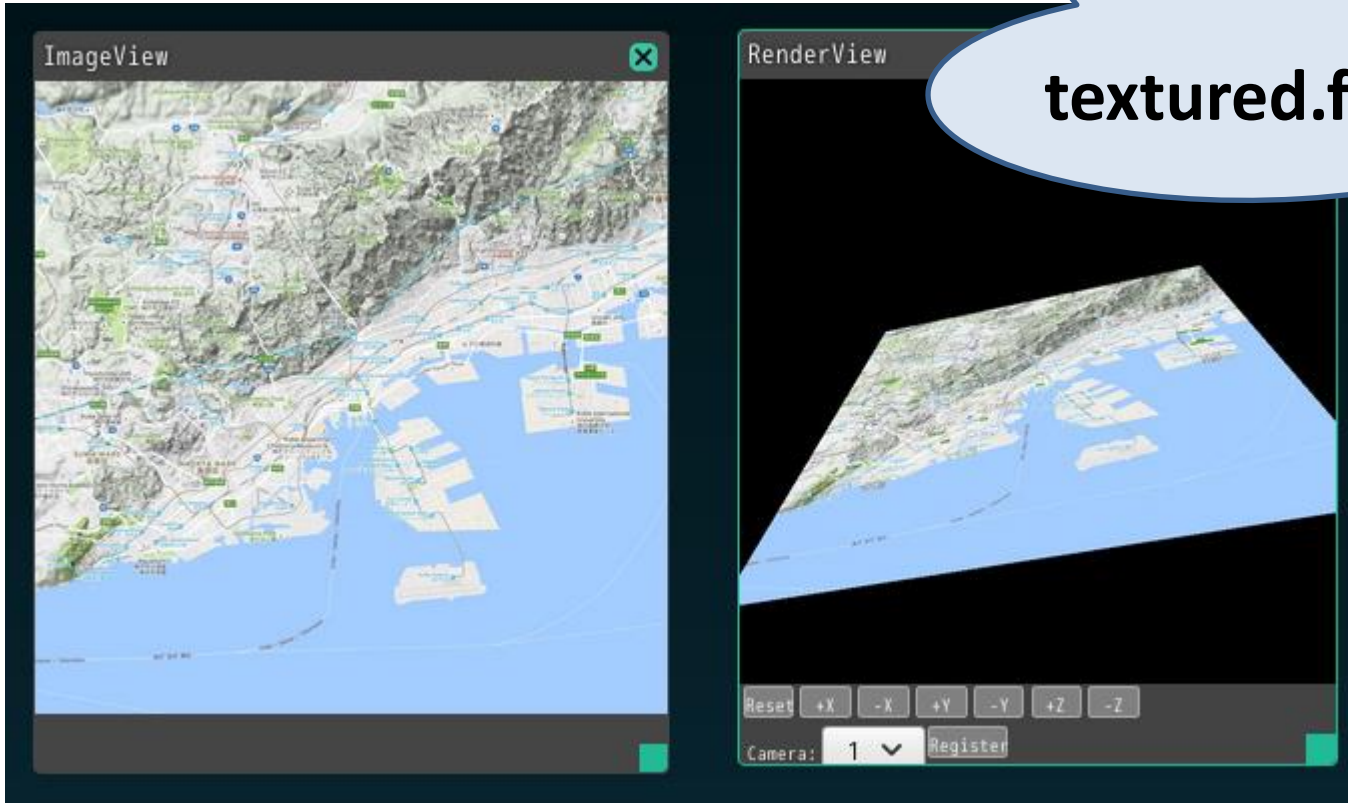
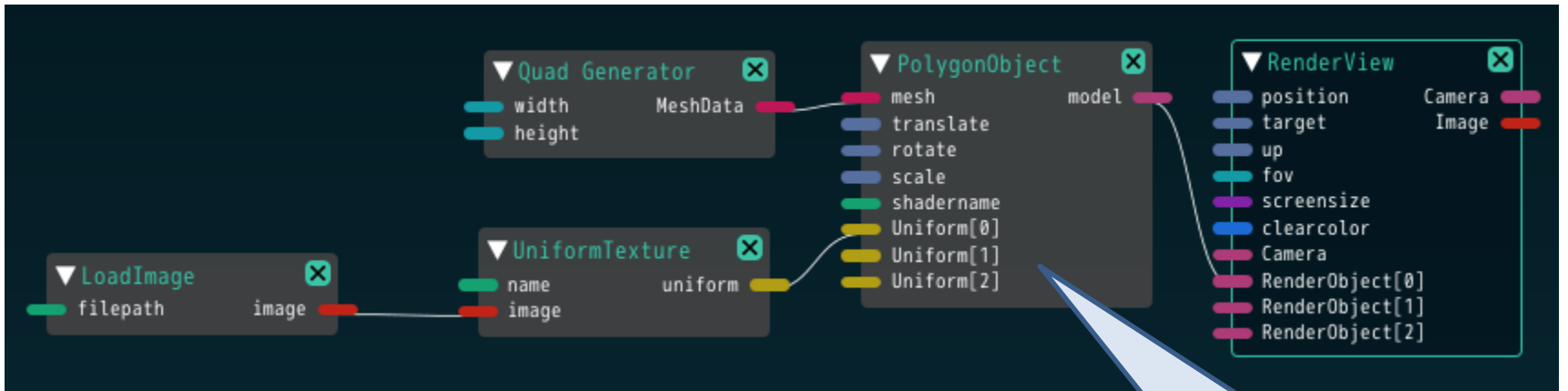


● PDB (Protein Data Bank) データ

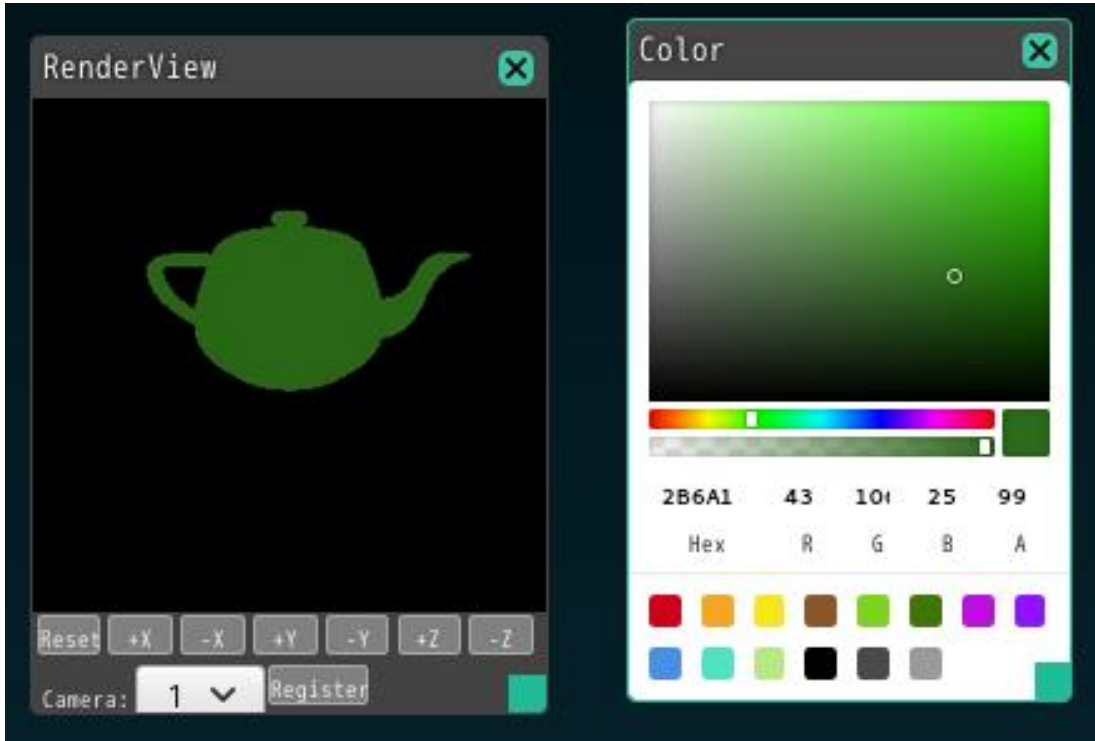
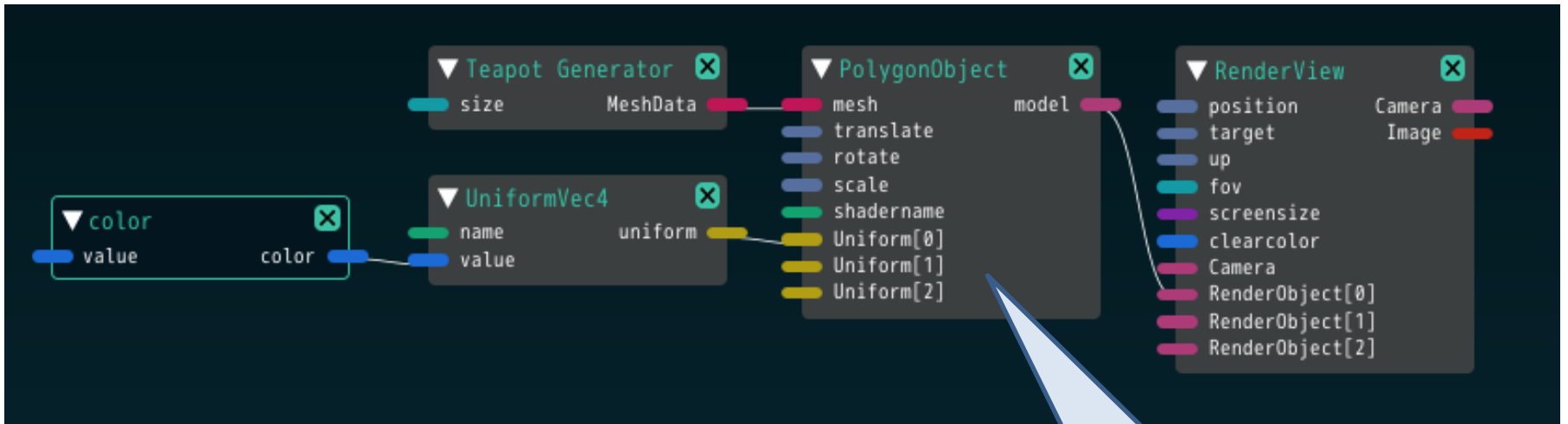


● 画像データ

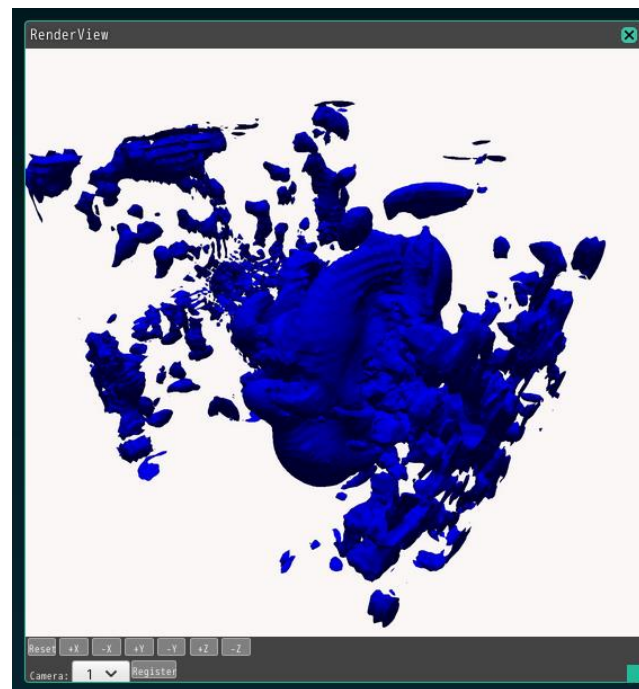
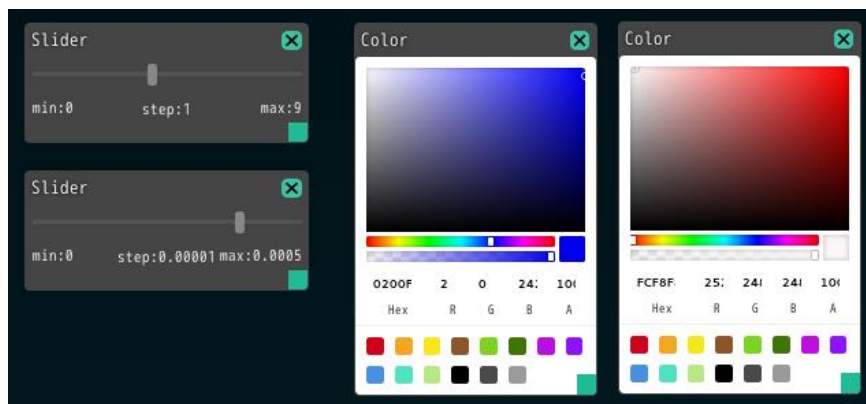
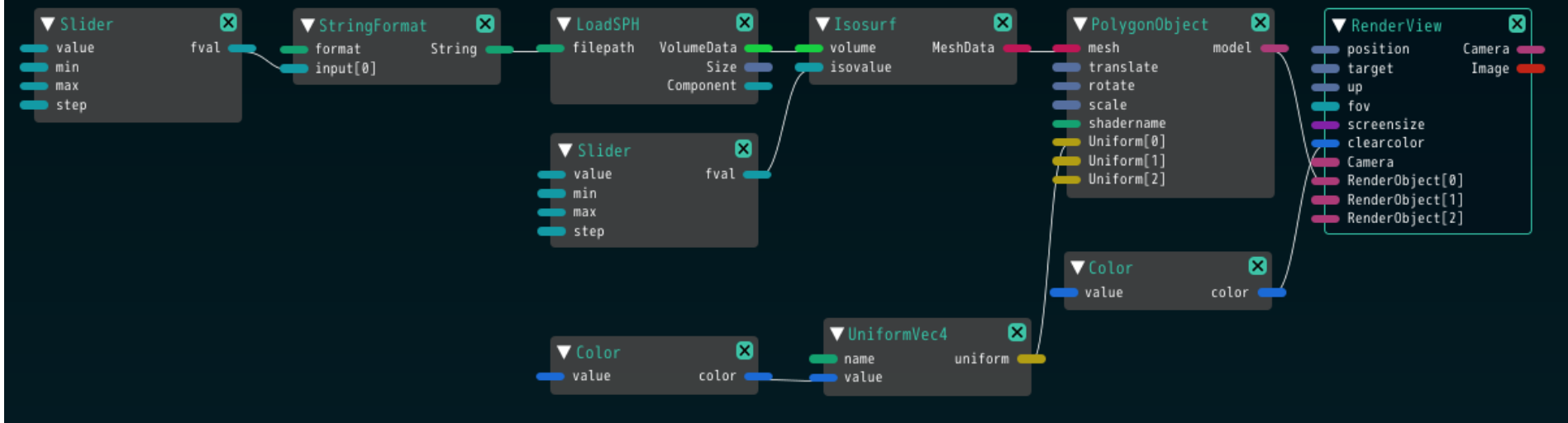




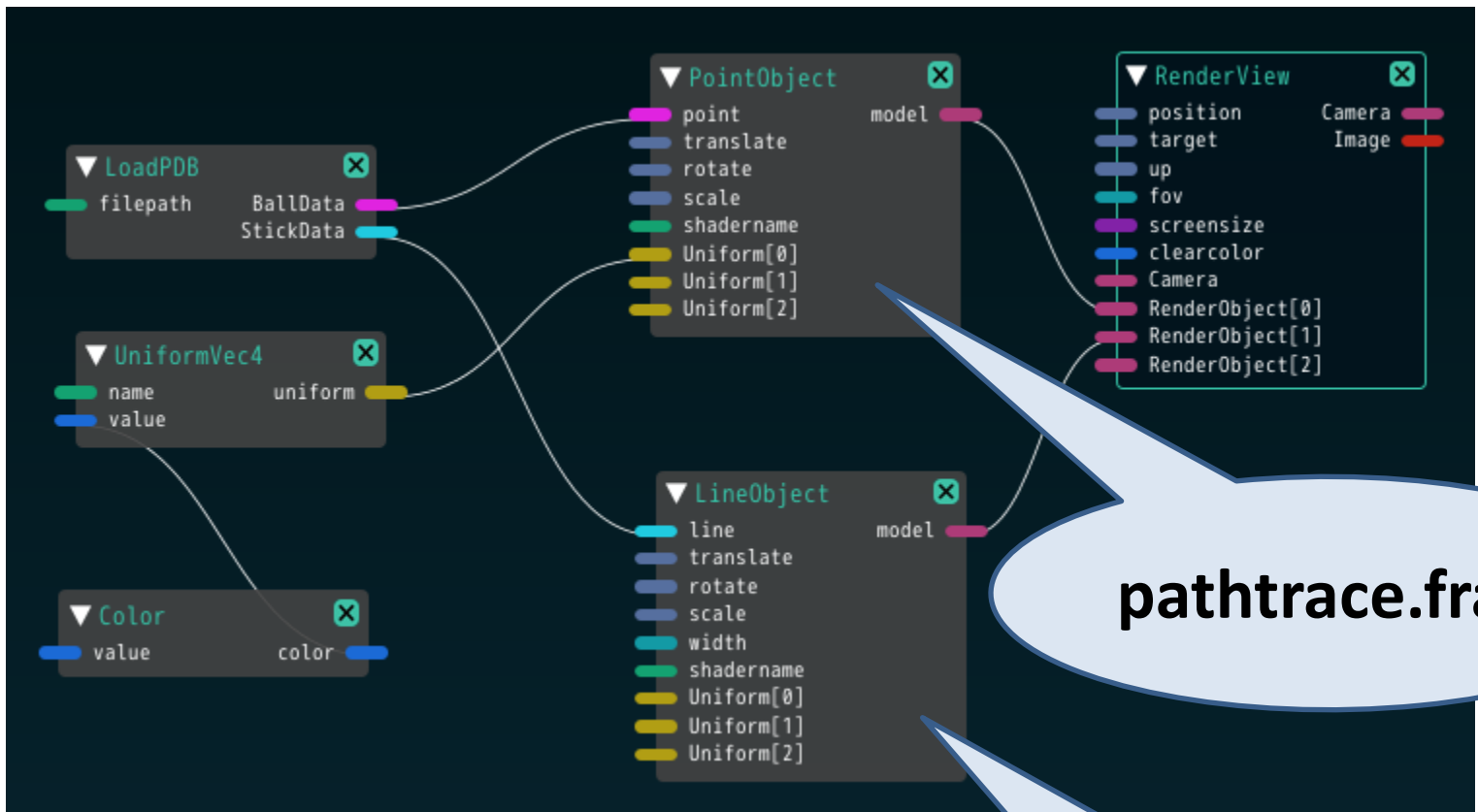
textured.frag



White.frag
↓
polygon.frag

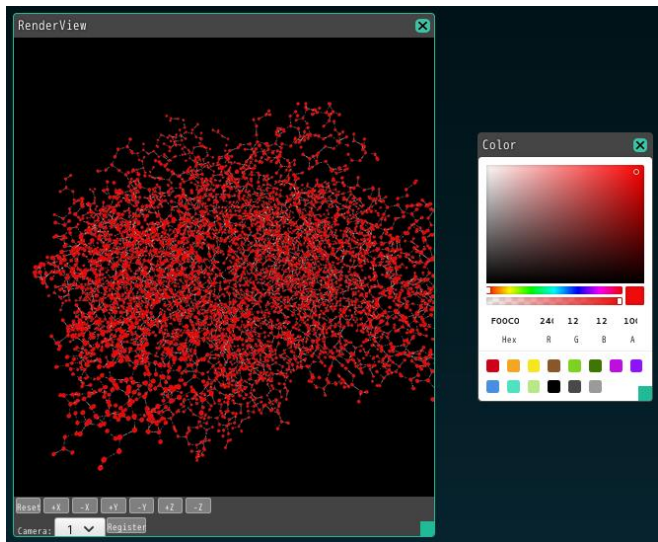


/home1/giud/share/HIVE/seminar_data/Sample/prs_tXXX.jpg

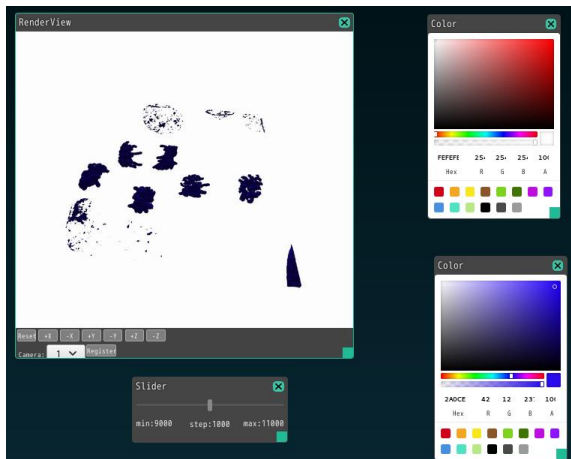
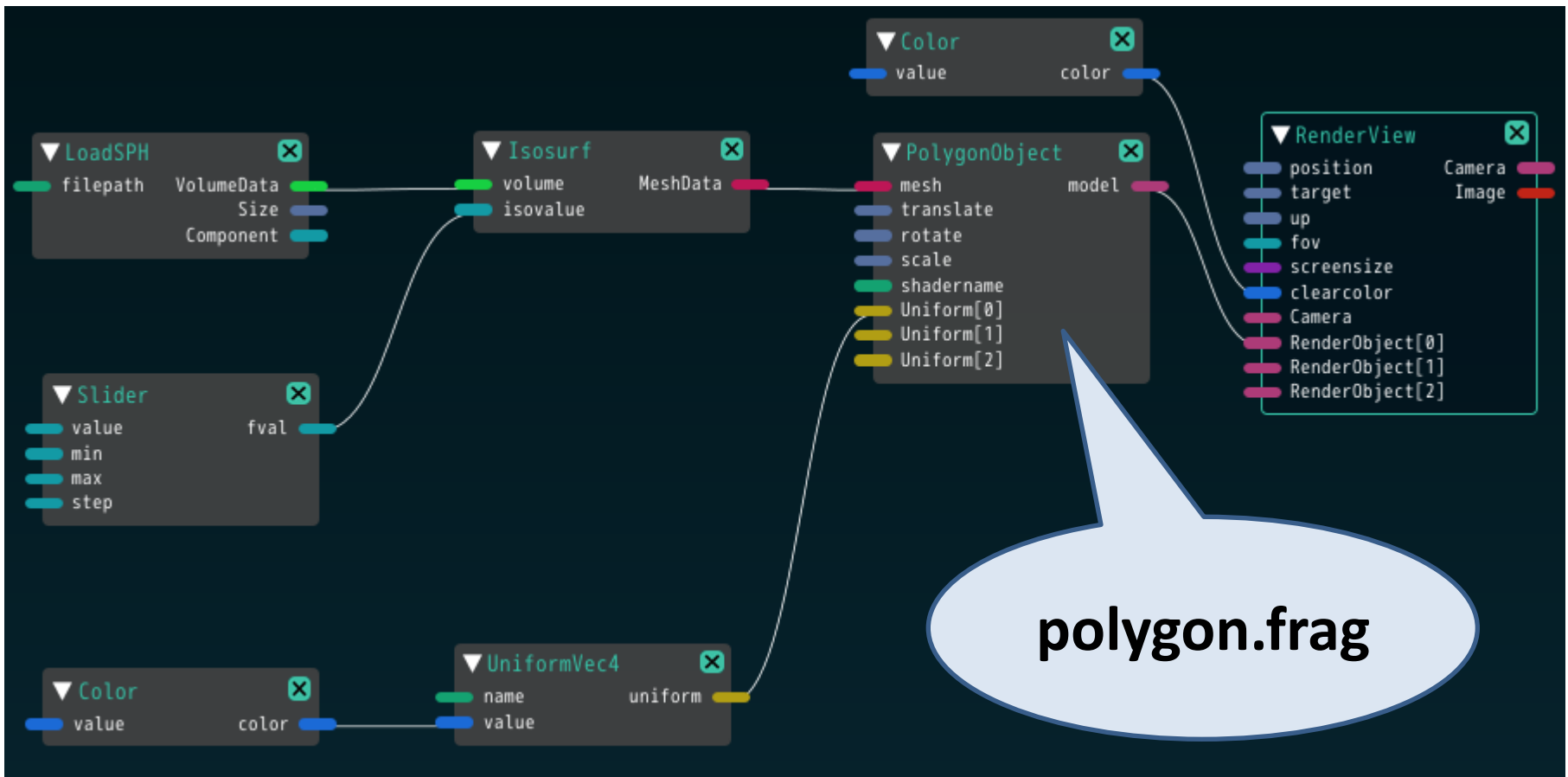


pathtrace.frag

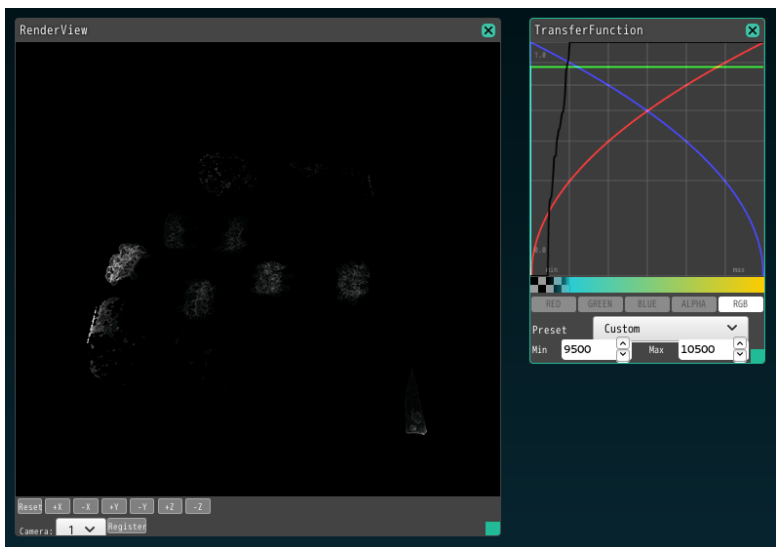
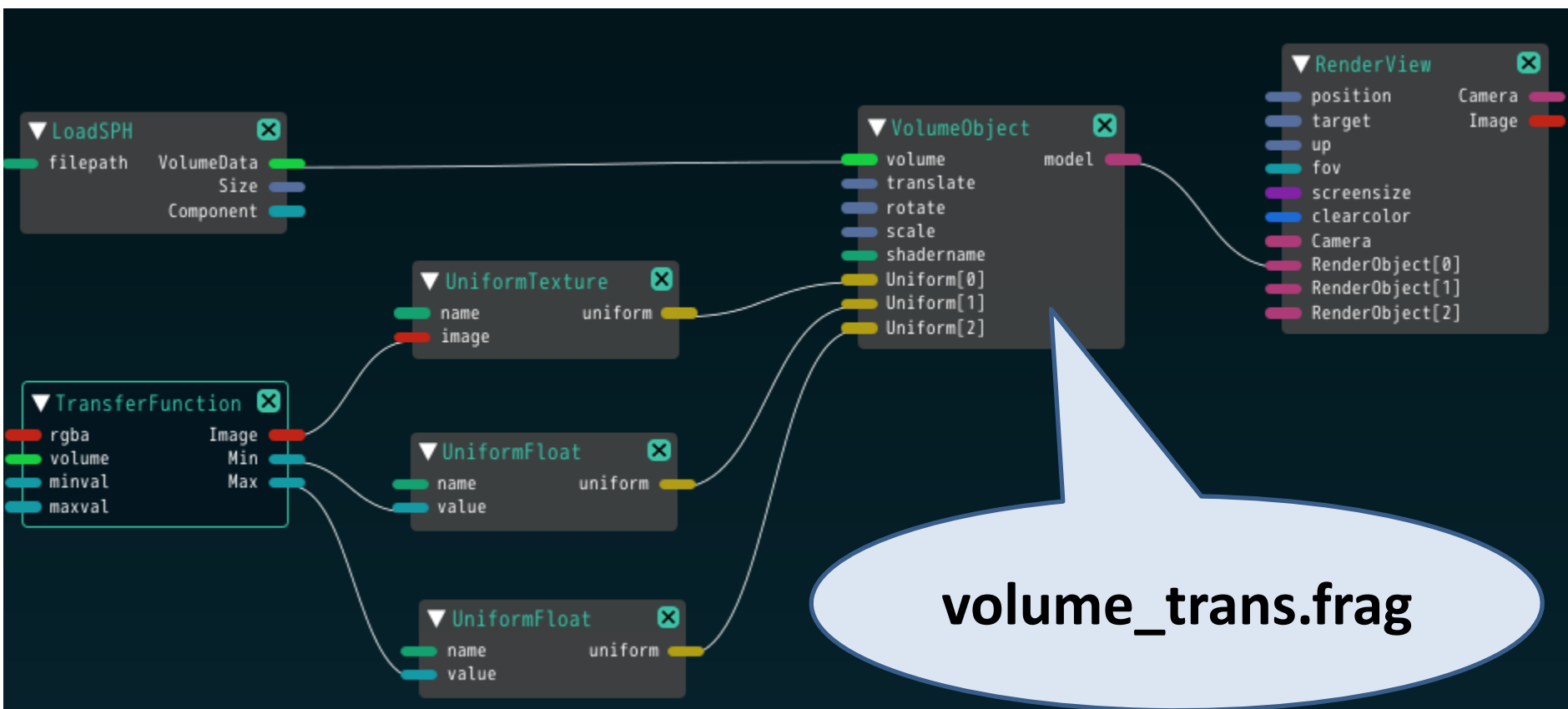
white.frag



/home1/giud/share/HIVE/seminar_data/Sample/4CL8.pdb



/home1/giud/share/HIVE/seminar_
data/CLST_Cell/cell.sph

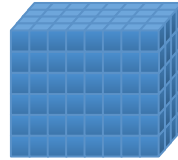


/home1/giud/share/HIVE/seminar_data/CLST_Cell/cell.sph

分割データ読み込み

● xDMlib (データマネージメントライブラリ) フォーマット

● Cartesian Data (CDMLib)



- Voxel (SPH, VOL, RAW)
- Structured (Plot3D XYZ, NetCDF SCALE)

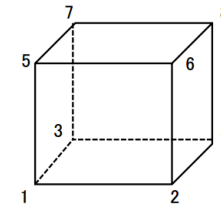
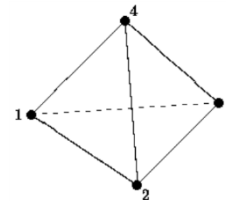
● Hierarchical Multi-Resolution Data (HDMLib)

- Structured (BCM, ParaView PVTI)



● Unstructured Data (UDMLib)

- CFD General Notation (CGNS)



● Particle Data (PDMLib)



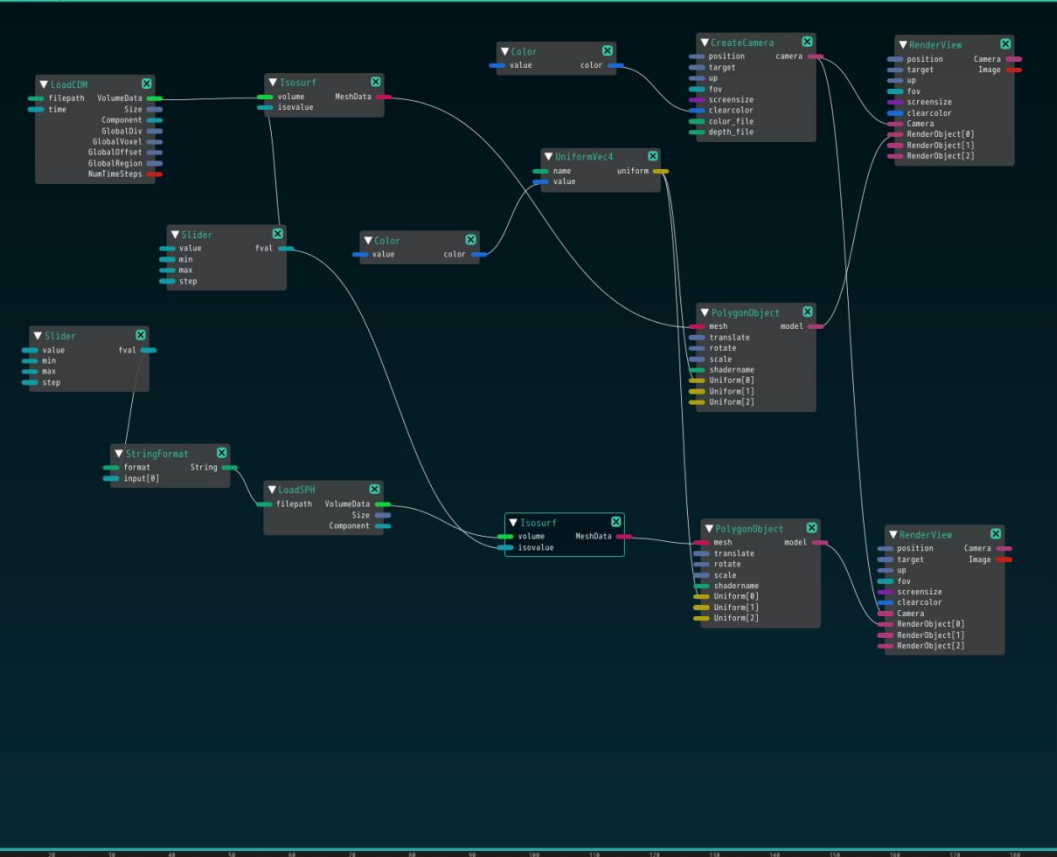
Module List

- LoadCDM
- Color
- CreateCamera
- Color Generator
- FloatToVec2
- FloatToVec3
- FloatToVec4
- Impetus
- Isosurf
- Line Generator

Property

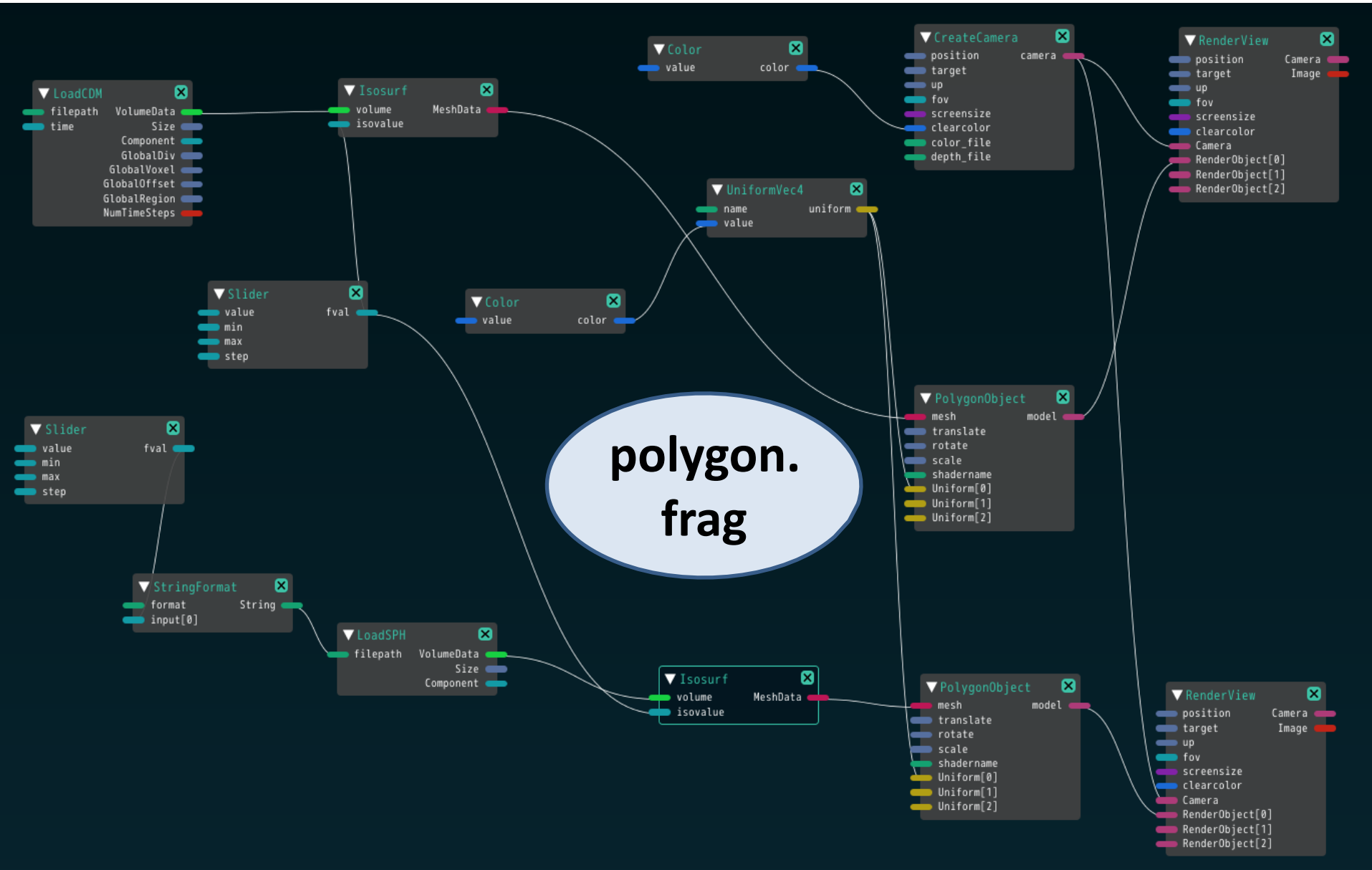
Isosurf

- label: isosurf
- volume: [Object]
- isovalue: 1



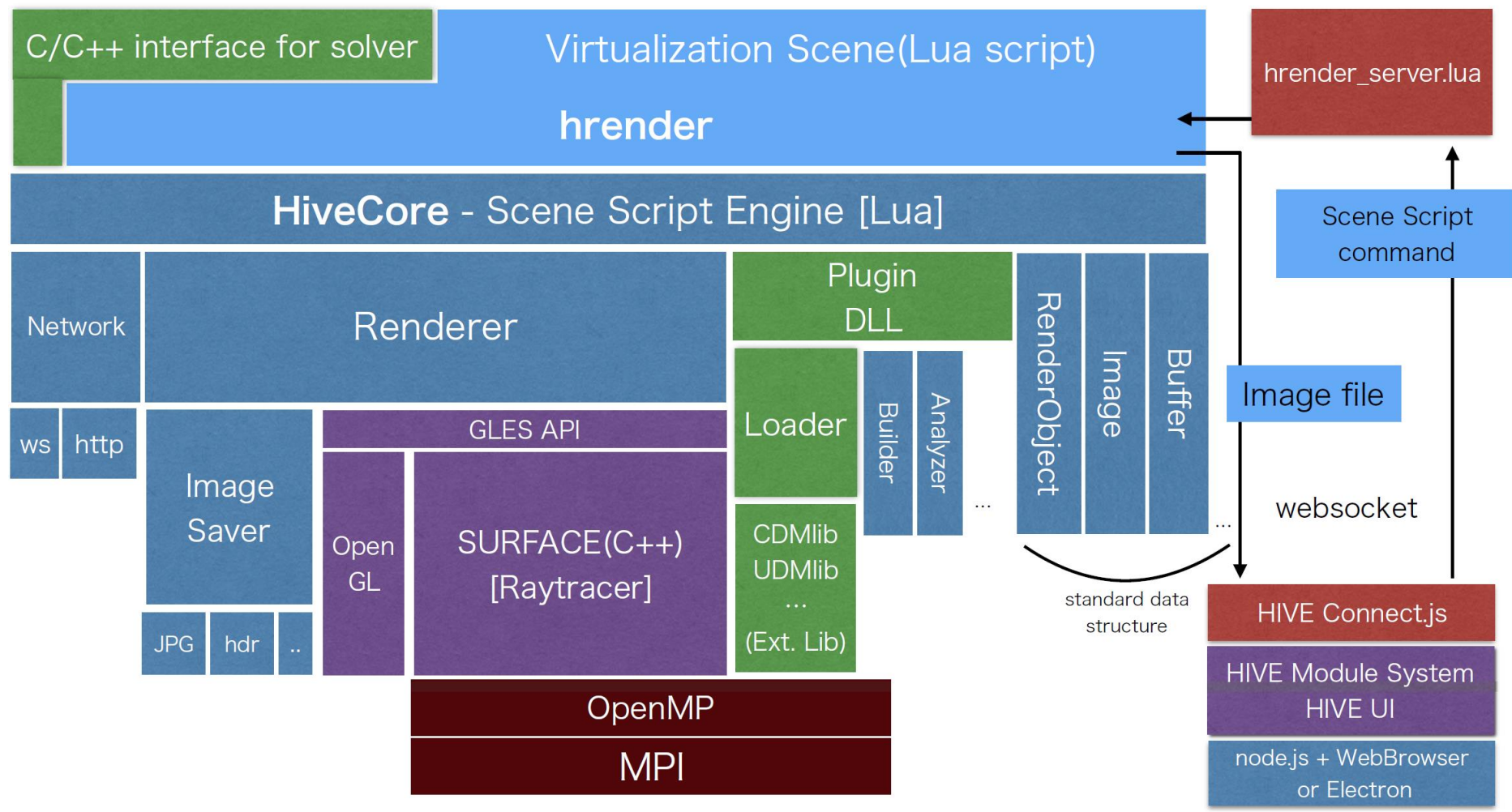
The right side of the interface features two **RenderView** windows and two control panels:

- Top RenderView:** Displays a 3D visualization of a complex, blue, fractal-like mesh structure. Below it is a **Slider** control with min:0, step:0.01, max:0.1, and a **Color** picker showing a blue color with hex code #2738D0.
- Bottom RenderView:** Displays a similar 3D visualization of the blue mesh structure. Below it is a **Slider** control with min:0, step:1, max:7, and a **Color** picker showing a red color with hex code #FA8E71.

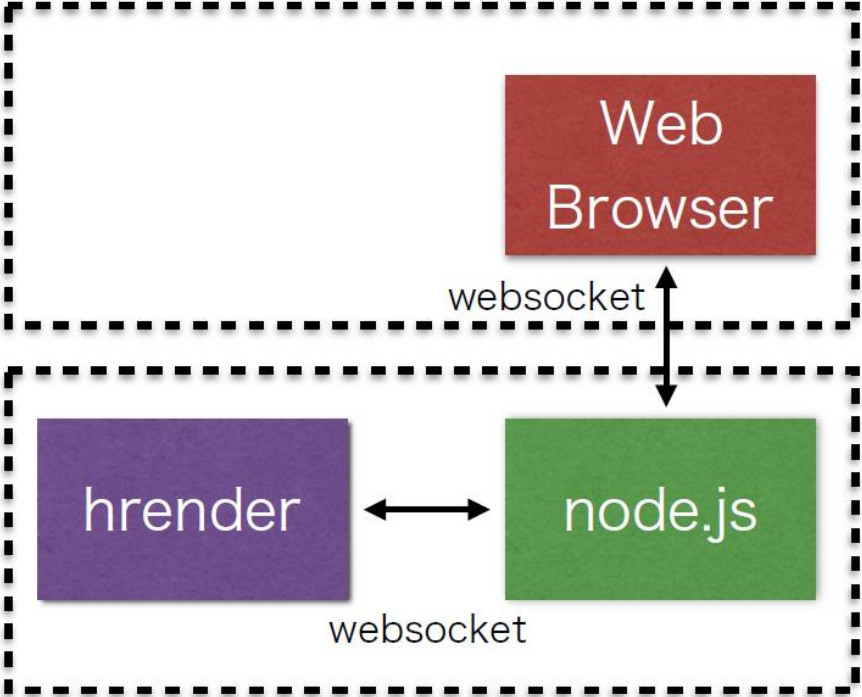


/home1/giud/share/HIVE/seminar_data/Sample/index_XXX.dfi

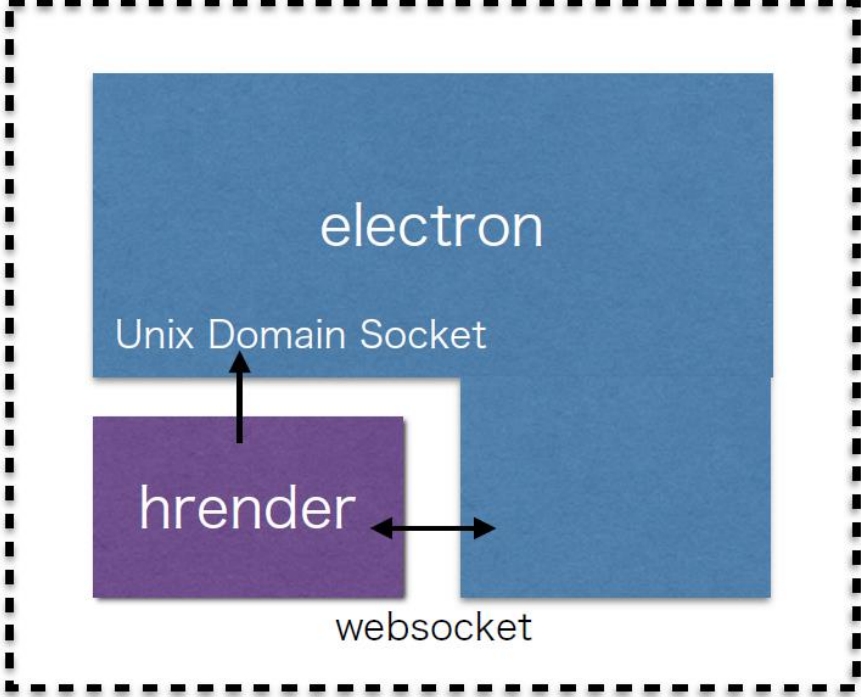
HIVE Architecture



Communication Protocol

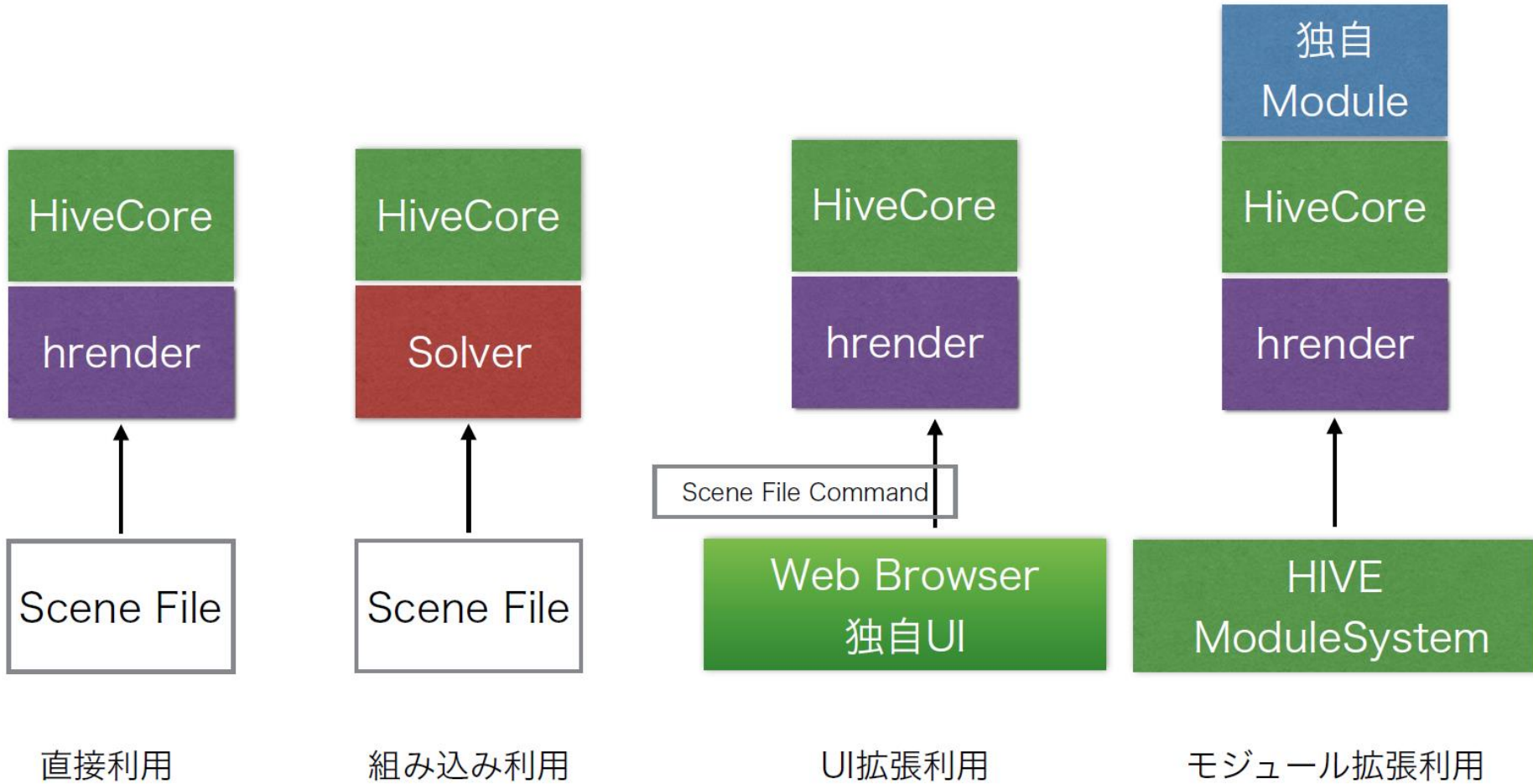


Server/Client
Websocket

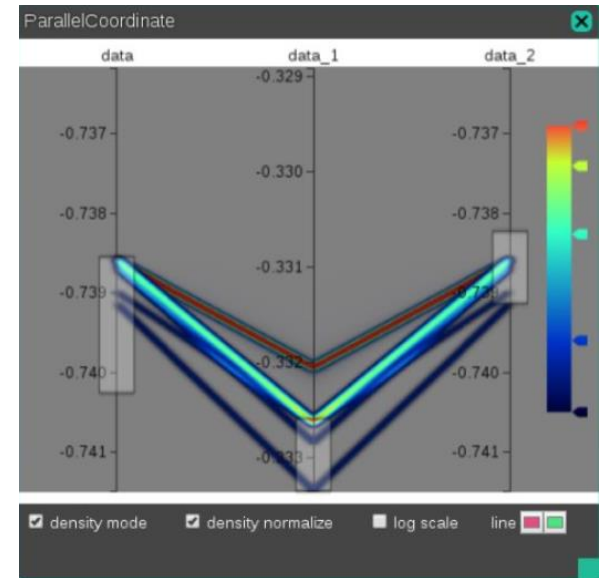
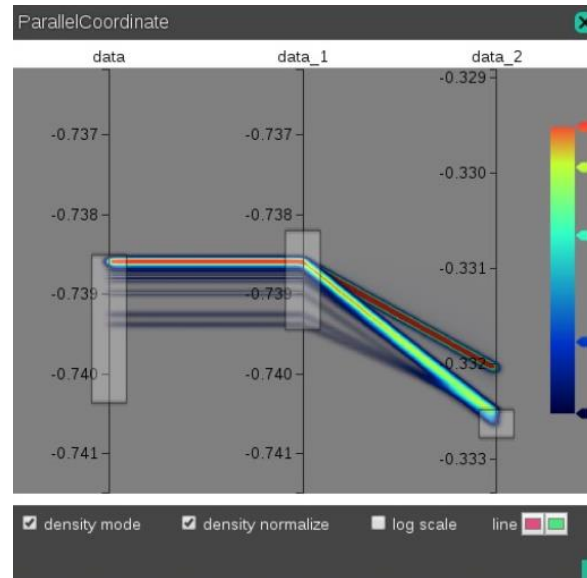
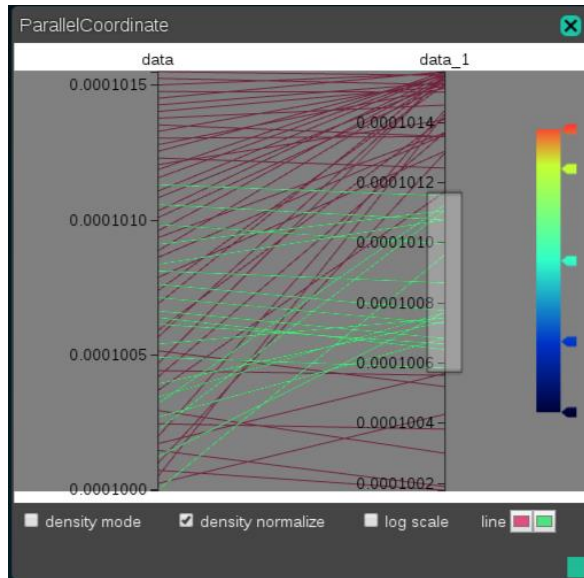


Local
Websocket
Unix Socket

拡張性



Parallel Coordinate



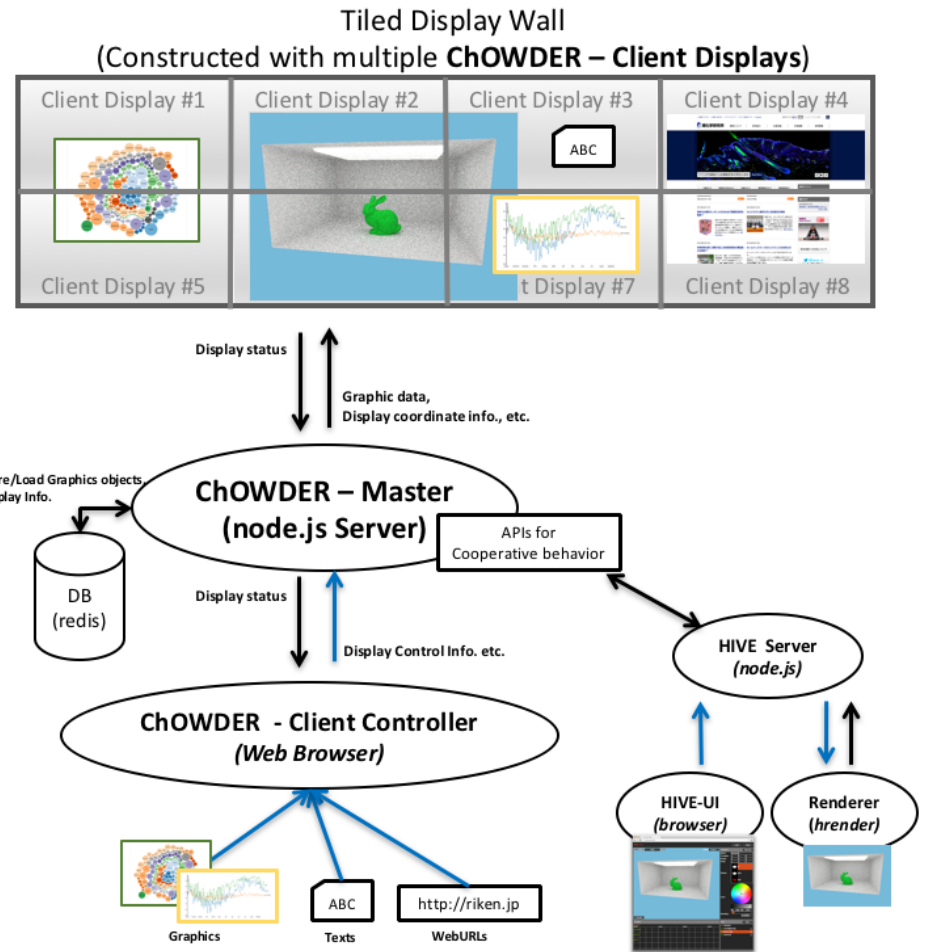
HIVEと連携するソフトウェアの紹介

- 可視化技術研究チームが開発主体となっている、ウェブブラウザベースのアプリケーション
 - ChOWDER
 - Collaborative Workspace Driver
 - web ベースのマルチプラットフォーム協調作業環境システム
 - HPC/PF
 - High Performance Computing Platform
 - 国プロ開発シミュレータやプリ・ポスト処理を統合的に利用する解析システム

ChOWDER



- シンプルな仕組みで大ピクセル空間を提供、チームでの共同作業・意思決定を支援するソフトウェアシステム
- 内閣府 SIP(戦略的イノベーション創造プログラム)/革新的設計生産技術プログラムから助成を受け開発中



ChOWDER

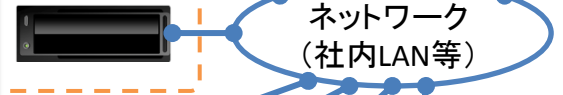
ChOWDERディスプレイ

- 複数並べることで大ピクセル空間(仮想ディスプレイ)を実現
- IE, Firefox, Chrome等一般的なウェブブラウザで動作
- ソフトウェアのインストール不要
- iPadやAndroidタブレットを並べて大画面にすることも可能



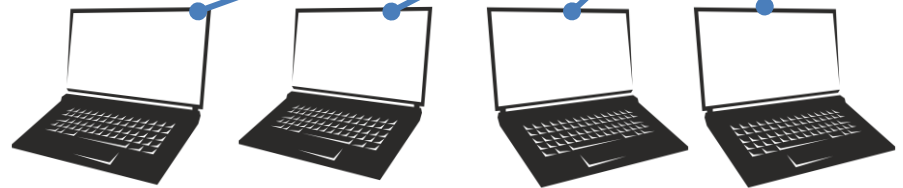
ChOWDERサーバ

- 仮想ディスプレイを管理
- Win, Linux, Macで動作
- ディスプレイあるいはコントローラのどれか1台がサーバを兼任してもよい
- 協調動作APIを提供。ChOWDER対応アプリ間での動作連携が可能



ChOWDERコントローラ

- 仮想ディスプレイ空間でのクライアントディスプレイの登録・位置を管理する
- 仮想ディスプレイ空間にコンテンツを登録・管理する
 - コンテンツ: 画像、テキスト、URL文字列(→Webページを表示)
- ウェブブラウザで動作。インストール不要



ChOWDER

ChOWDERディスプレイ

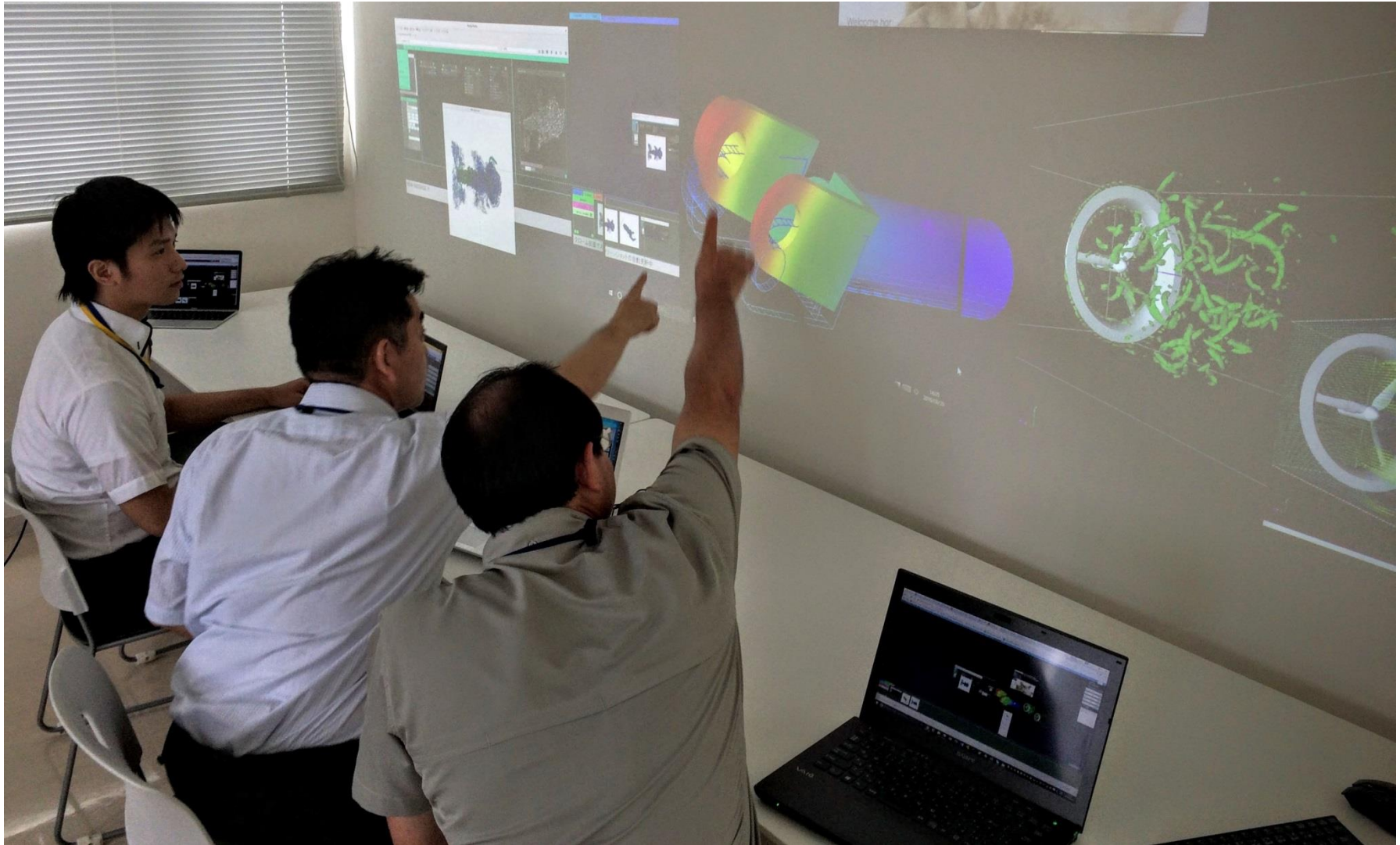


ChOWDERサーバ

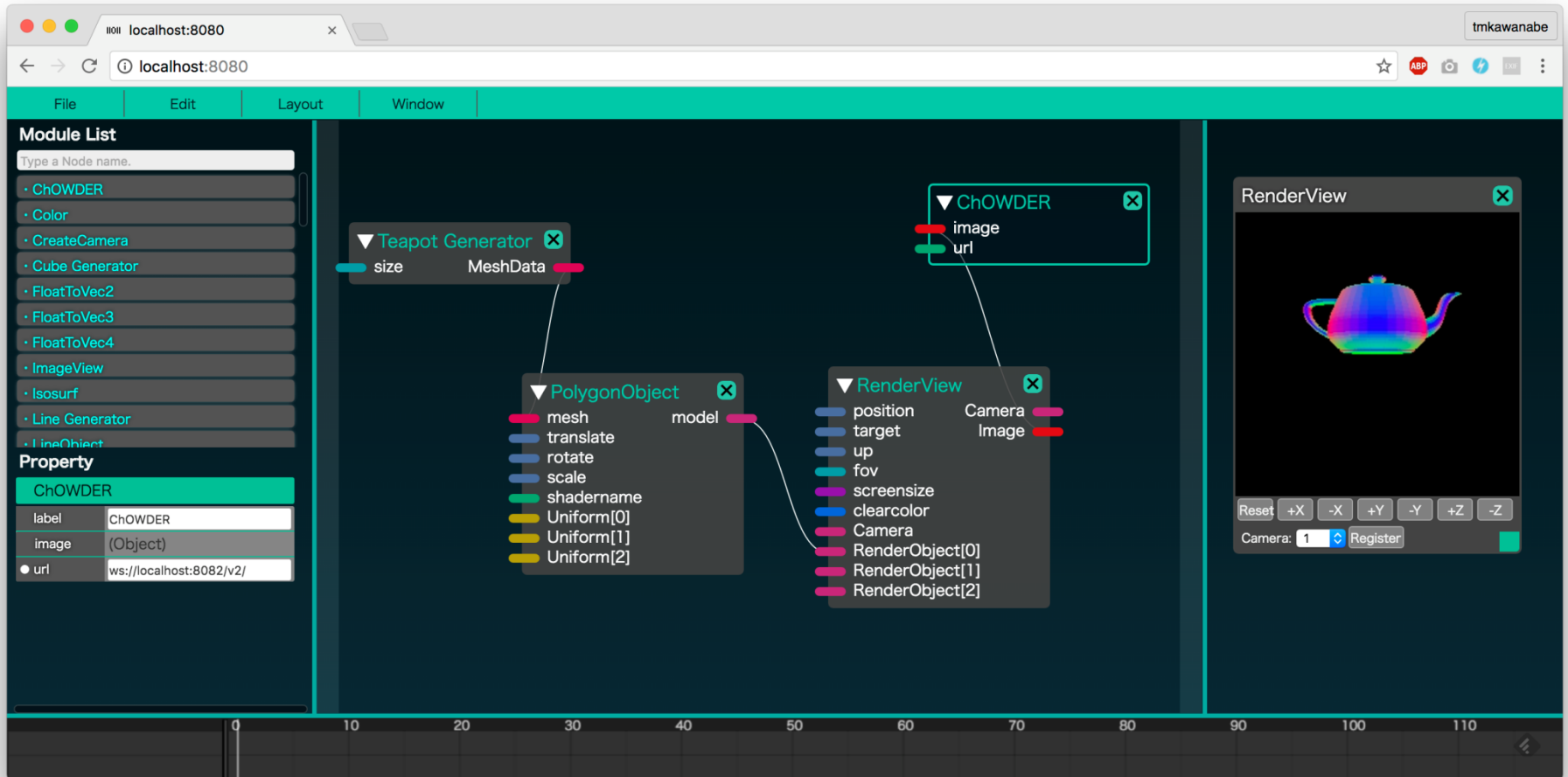
ChOWDERコントローラ



ChOWDER



HIVE-ChOWDER連携



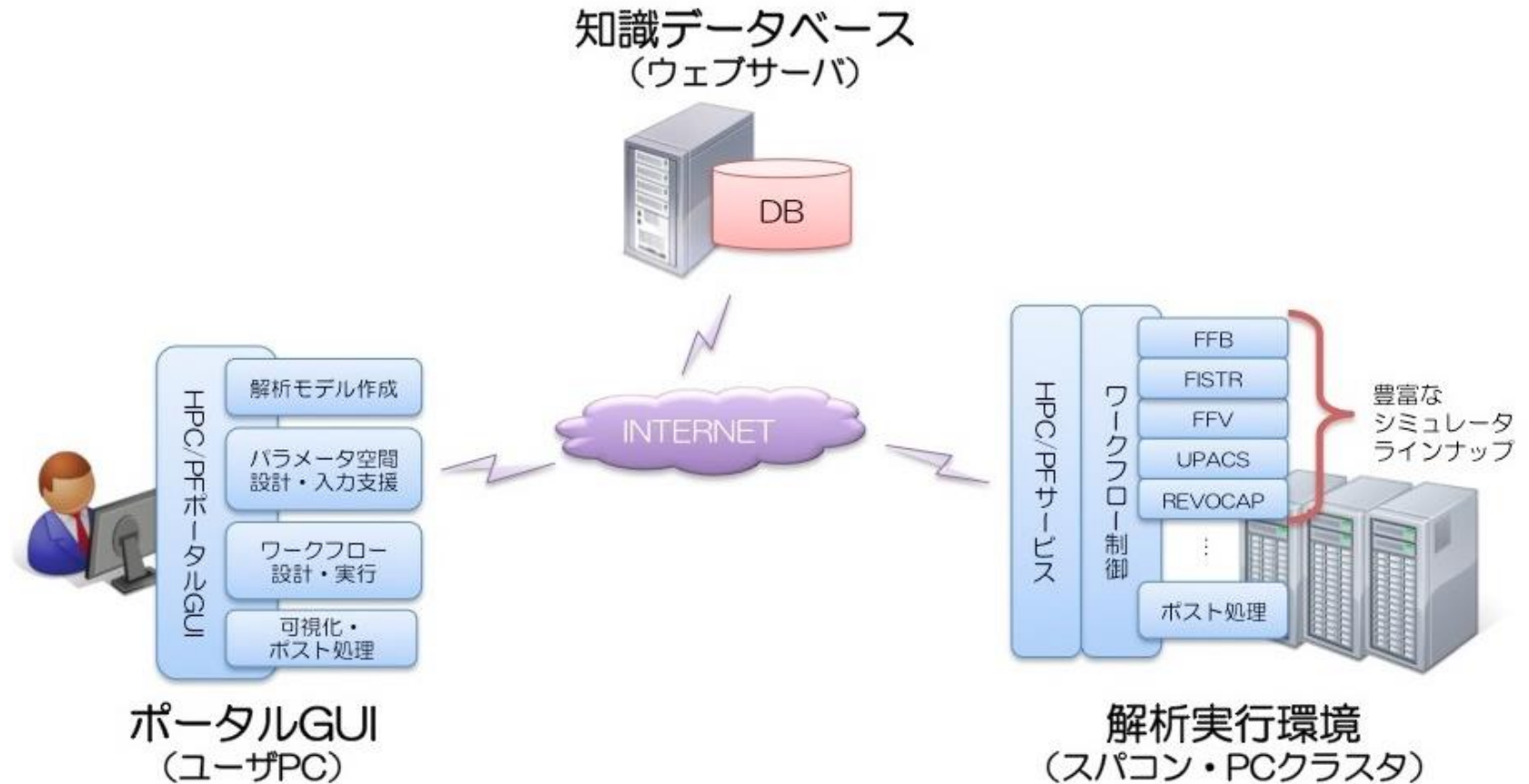
HPC/PF

- 国プロ開発の各種シミュレータとプリ・ソフトを統合的に利用できる解析プラットフォーム
- GUIはウェブブラウザ上で動作
 - HIVE同様、ノードを繋いでワークフローを組み立てる

The screenshot displays the HPC/PF Project Editor web interface. The browser address bar shows the URL: localhost:8080/editor.html?%2FUsers%2Fakinaga%2FDocuments%2FProjects%2Fhpcpf-archive%2FPHASE_PtCrystal. The interface includes a top navigation bar with buttons for 'Project Editor', 'Workflow', 'Info', 'Log', 'Host', and 'Vim'. Below the navigation bar, the current project path is shown as /Users/akinaga/Documents/Projects/hpcpf-archive/PHASE_PtCrystal/case_post/. The main workspace is divided into a file explorer on the left and a workflow editor on the right. The file explorer shows a list of files: case_post, case_solver_bcc, case_solver_fcc, nodedata.json, pmd.json, pwf.lua, and pwf_20160108052504.lua. The workflow editor shows a project named 'PHASE parameter study' with three nodes: case_solver_bcc, case_solver_fcc, and case_post. Each node is a green box containing its status (Finished) and associated data (e.g., energy, number of threads). Red arrows indicate the flow of data between nodes, showing that case_solver_bcc and case_solver_fcc feed into case_post. A 'property' sidebar is visible on the right side of the workflow editor.

HPC/PF

- 3つのサブシステムで構成



HPC/PF 知識DBコンテンツ

計算工学ナビ [Knowledge Base] 3 0 + 新規 投稿の編集 こんにちは、tkwanab さん!

計算工学ナビ

レポート ニュース 解析事例データベース ライブラリ Q&A コミュニティ

円柱周りの流れの解析

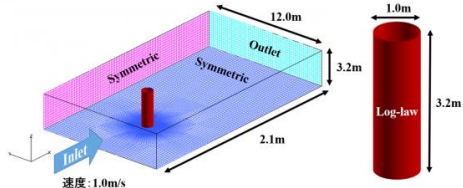
● ソフトウェア
ものづくり > 流体解析 > FFR

● V&V区分
Verification 検証

● HPC/PF再現実行対応事例

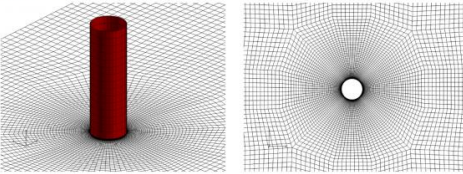
概要
本例題では、円柱周りの流れの解析を行い、円柱後方の渦列を確認した。

解析モデル
解析空間内部に直径1.0mの円柱モデルを配置した。入口には流入条件として流速1.0m/s、出口には流出条件を設定した。四方の壁には周期境界条件を与えた。円柱の表面は対称条件である。



使用ソフトウェアモジュール
FrontFlow/red-v3.1

計算格子
円柱近傍の格子の様子を下図に示す。総要素数はおよそ15万要素である。



▶ V & V 区分
性能測定 (7)
Verification 検証 (79)
Validation 妥当性確認 (22)
Application 応用 (12)

▶ ソフトウェア
ものづくり (90)
・OpenFOAM (16)
・OpenFOAM Version 2.2.2 (16)
・構造解析 (32)
・ADVENTURECluster Solver (3)
・Front ISTR (29)
・流体解析 (42)
・FFR (2)
・FFV (13)
・FrontFlow/blue (27)
・複合材料強度信頼性評価 (9)
・FrontCOMP (9)
・音響解析 (2)
・FFB-Acoustics (2)
ナノテクノロジー (61)
・第一原理電子状態解析 (61)
・PHASE (61)
ライフサイエンス (33)
・フラグメント分子軌道法 (13)
・ABINIT-MP (13)
・標準密度汎関数法 (20)
・ProteinDF (20)

FrontFlow/red の例

計算工学ナビ

レポート ニュース 解析事例データベース ライブラリ Q&A コミュニティ

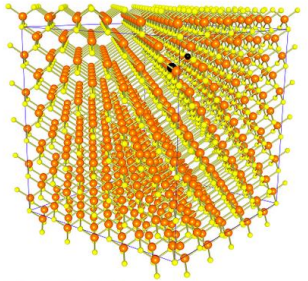
ナノテクノロジー 第一原理電子状態解析ソフトウェア PHASE の性能評価

● ソフトウェア
ナノテクノロジー > 第一原理電子状態解析 > PHASE

● V&V区分
性能測定

概要
PHASE 通常版 (2軸分割, PHASE-2D) と大規模並列版 (3軸分割, PHASE-3D) の並列処理性能を評価した。計算機は、FOCUSスパコンのEシステムを利用した。PHASE-2Dは、フラットMPI並列で、PHASE-3Dは、フラットMPI及びOpenMP-MPIハイブリッド並列で測定した。PHASE-3Dの性能は非常に高く、400コアまで良好にスケールした。また、400コアを超える規模ではハイブリッド並列が有利になると予想される。

解析モデル
4H-SiC中の欠陥構造。1000原子からなる立方晶格子 (青線で示す) からSiとCをひとつずつ抜いた。Siを青色で、Cを黄色で、抜き出したSiとCは黒色で表す。



使用ソフトウェアモジュール
PHASE/0 2014.00 (公開前版) を使用。

計算条件
以下に記す全ての計算条件に対して、SCF計算を10回繰り返すジョブを実行し1回あたりの経過時間を評価した。

● PHASE-2D フラットMPI並列

	20	20	20	40	40	100	200	400
#cores	20	20	20	40	40	100	200	400
#nodes	1	2	4	2	4	5	10	20
#task/node	20	10	5	20	10	20	20	20
npes	20	20	20	40	20	100	200	400
ne	20	20	20	40	20	100	200	400
nk	1	1	1	1	1	1	1	1

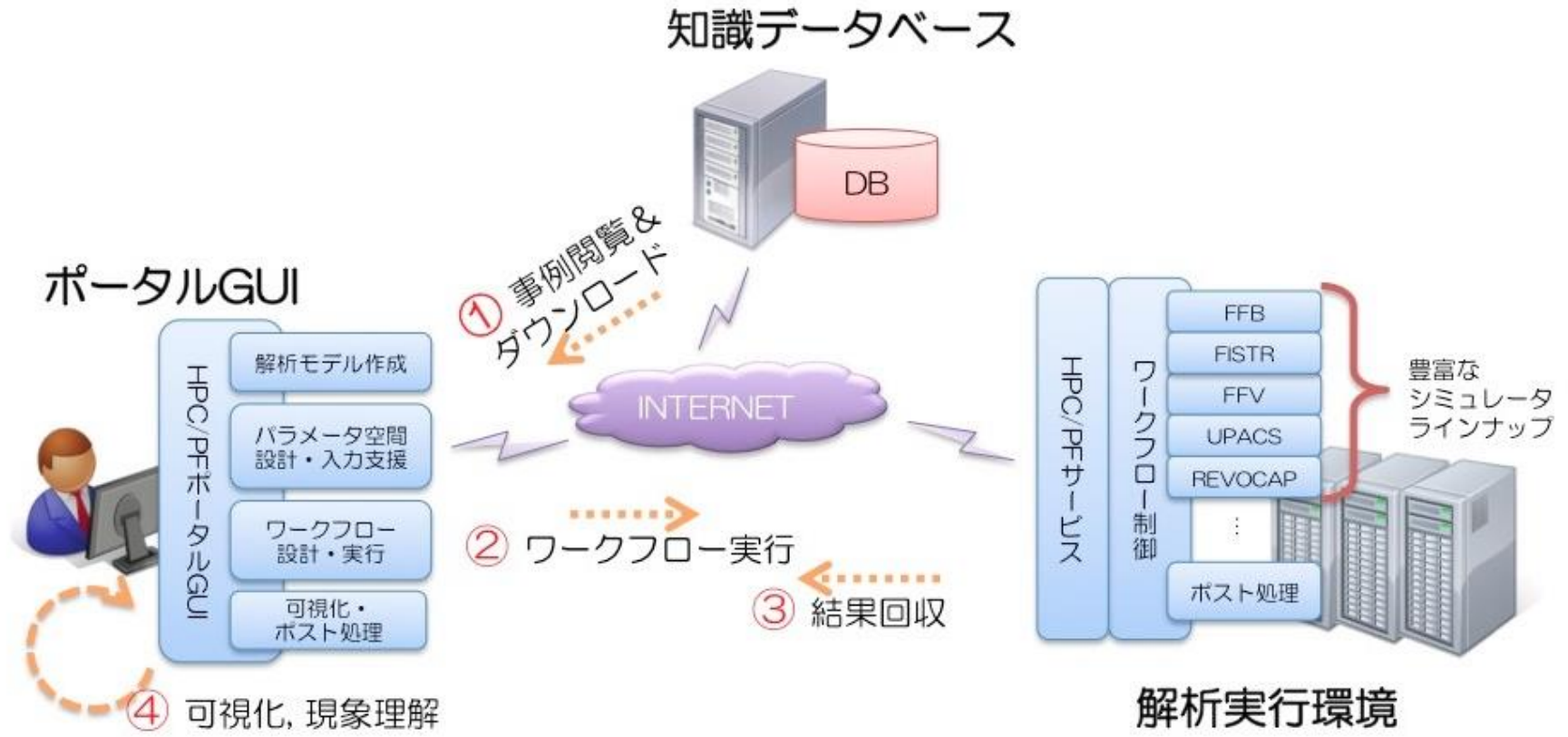
ne, nikiは、2軸分割のパラメータ。#task/node=20は1ノードの全20コアを使用、途中

▶ V & V 区分
性能測定 (7)
Verification 検証 (85)
Validation 妥当性確認 (22)
Application 応用 (12)

▶ ソフトウェア
ものづくり (103)
・OpenFOAM (16)
・OpenFOAM Version 2.2.2 (16)
・最適化設計 (2)
・CHEETAH (2)
・構造解析 (32)
・ADVENTURECluster Solver (3)
・Front ISTR (29)
・流体解析 (53)
・FFR (4)
・FFV (15)
・FrontFlow/blue (28)
・GKV (2)
・GT5D (2)
・LANS3D (3)
・複合材料強度信頼性評価 (10)
・FrontCOMP (10)
・音響解析 (2)
・FFB-Acoustics (2)
ナノテクノロジー (65)
・第一原理電子状態解析 (65)
・PHASE (65)
ライフサイエンス (36)
・フラグメント分子軌道法 (16)
・ABINIT-MP (16)
・標準密度汎関数法 (20)
・ProteinDF (20)

PHASEの例

HPC/PF 知識DB解析事例の追体験

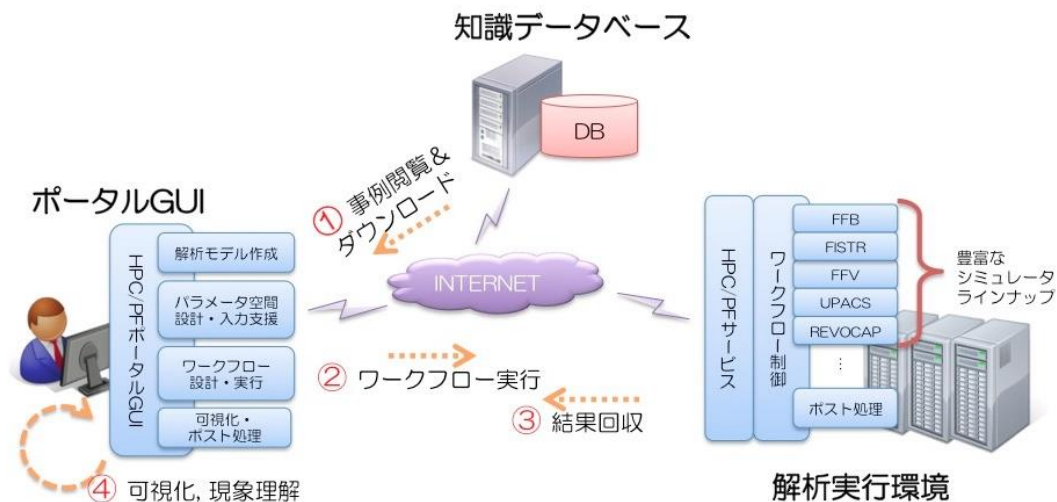


HPC/PF利用のメリット

HPC/PF-GUIを使うことで、

- PC→スパコンへのファイル転送
- ジョブスケジューラへのジョブ投入・監視
- ジョブ終了後の解析結果ファイル回収

といった、**解析処理の本質と関係のない**が手間のかかる**手順**をHPC/PF-GUIが自動実行します。



ソフトウェアの配布

- **HIVE**

- <http://avr-aics-riken.github.io/HIVE/>

- **ChOWDER**

- <https://github.com/SIPupstreamDesign/ChOWDER>

- **HPC/PF**

- <https://github.com/avr-aics-riken/hpcpfGUI>