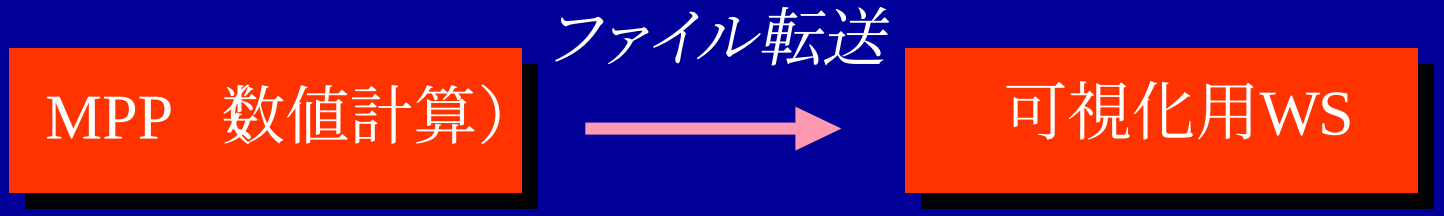
超並列計算機のための並列可視化システム

計算物理学研究センター　板倉 憲一
（アーキテクチャ研究室
[現在　農業生物資源研究センター]　沼 寿隆）

研究の目的

- 従来の一般的な方法



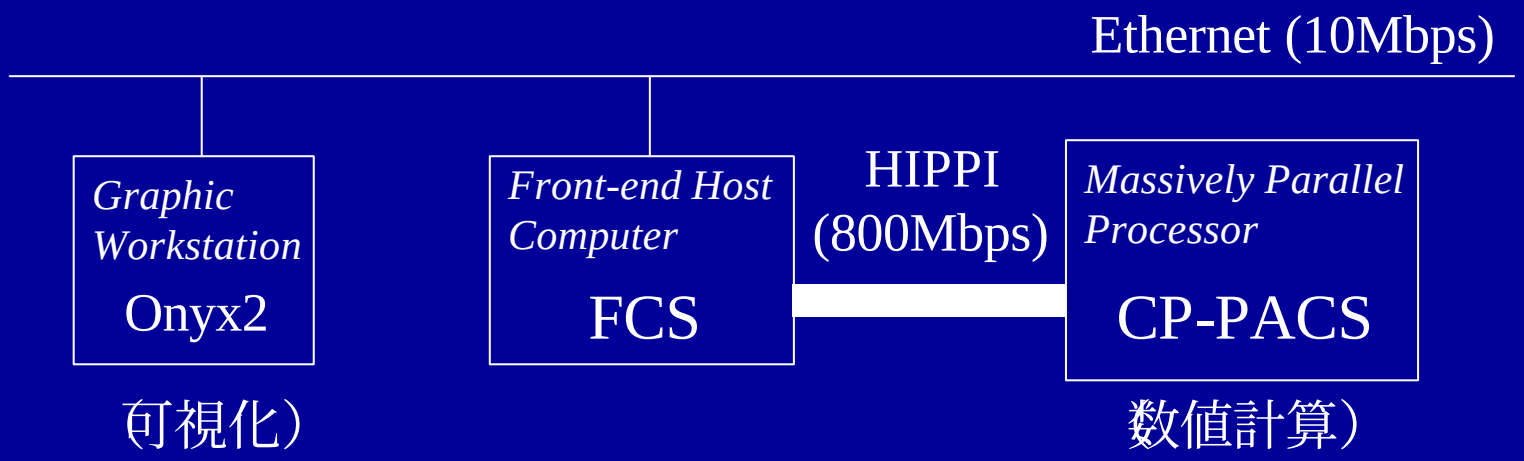
問題点

  - 一括ファイル転送、一括可視化のコスト大
  - 計算の妥当性をリアルタイムに検証できない

- **数値計算と可視化を同時実行させる並列可視化システムの開発**

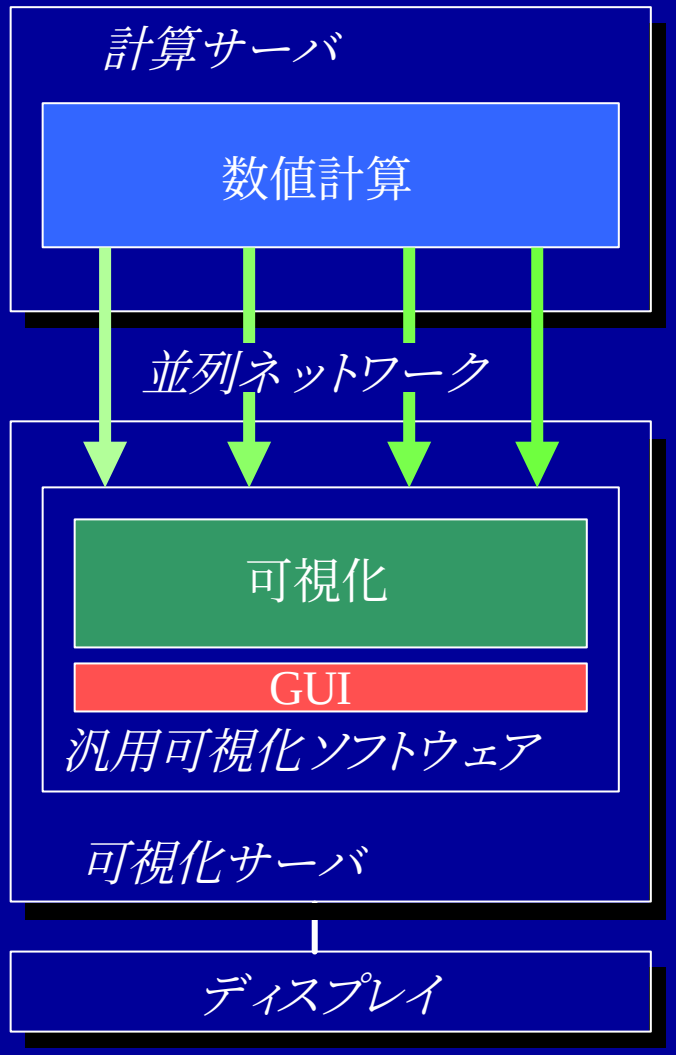
  - 実験環境
    - 超並列計算機 CP-PACS　数値計算）
    - SGI Onyx2, Origin2000　可視化）

これまでの可視化手順とその問題点



- 途中経過をモニターできない
- ファイル転送や可視化処理に時間がかかる
- 並列に計算したデータを一度ひとつに束ねる逐次化がオーバヘッドになる

実装方針



- 高スループットなネットワーク
- 大規模可視化データを処理する可視化サーバ
- データ転送および可視化処理もできる限り並列化
- 汎用可視化ソフトウェア（MVE）を利用

ハードウェア構成

数値計算サーバ
超並列計算機 CP-PACS
（2048PU、128IOU）

Switching
HUB

×8

×8

100Base-TX
Ethernet
による並列接続

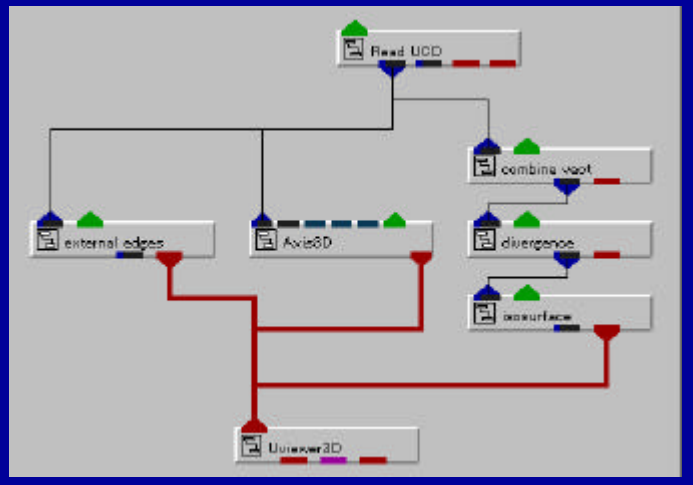
並列入出力システム

可視化サーバ
SGI Onyx2（4Proc.）

ファイルサーバ
SGI Origin2000（8Proc.）

MVE (Modular Visualization Environment)

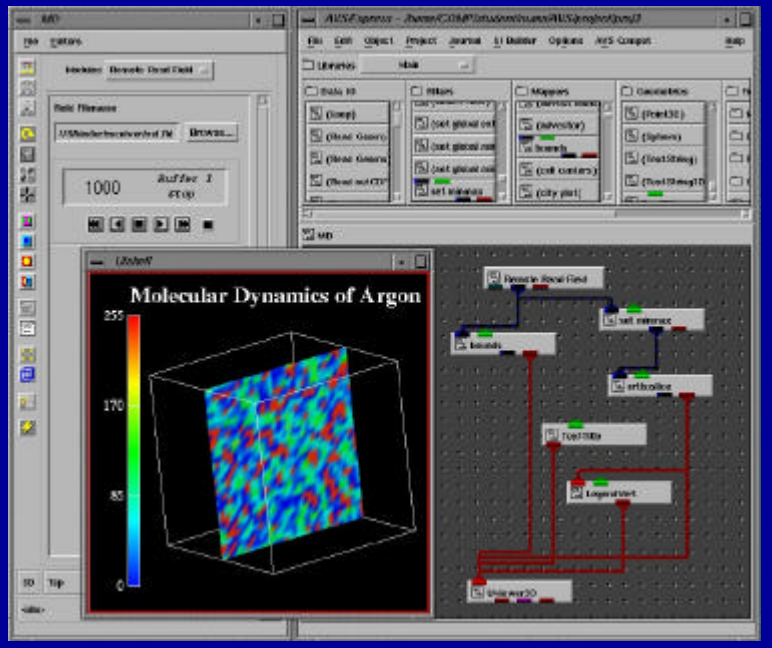
- モジュールを繋ぎ合わせて
  ネットワークを作ることで、一連の
  可視化処理を行なうプログラミング
  環境
  - 多数の標準モジュールが用意されて
    いる
  - ユーザ独自のモジュールを開発し、
    既存のモジュール群に組み込むことが
    可能
  - 汎用性と再利用性
  - 例：AVS/Express、Khoros、Iris Explorer、…



- **可視化処理部分とフロントエンドGUIはAVS/Expressを基本に構築する。**

可視化システムの機能

- すべての操作をフロントエンドのGUIから行なう
  - 数値計算の実行開始、可視化パラメータの設定・変更 など
- 可視化方法の動的切り替え
  - シミュレーションプログラムを止めることなく、可視化のパラメータや可視化手法を設定・変更できる
- 一時停止・再実行機能
  - 停止、巻き戻し（早送り）
- 時系列データバッファリング
  - 過去のデータに遡り、新たな視点や方法でデータを解析できる
- 可視化パラメータの保存



並列入出力システム（PIO）

- 並列ネットワークを有効に使うためには …
  - チャネルの本数を考慮したプログラミング
    → 16本ものチャネルを使用するプログラミング
  - チャネルの負荷分散
    → ファイル転送ではないためデータ量が予想できない

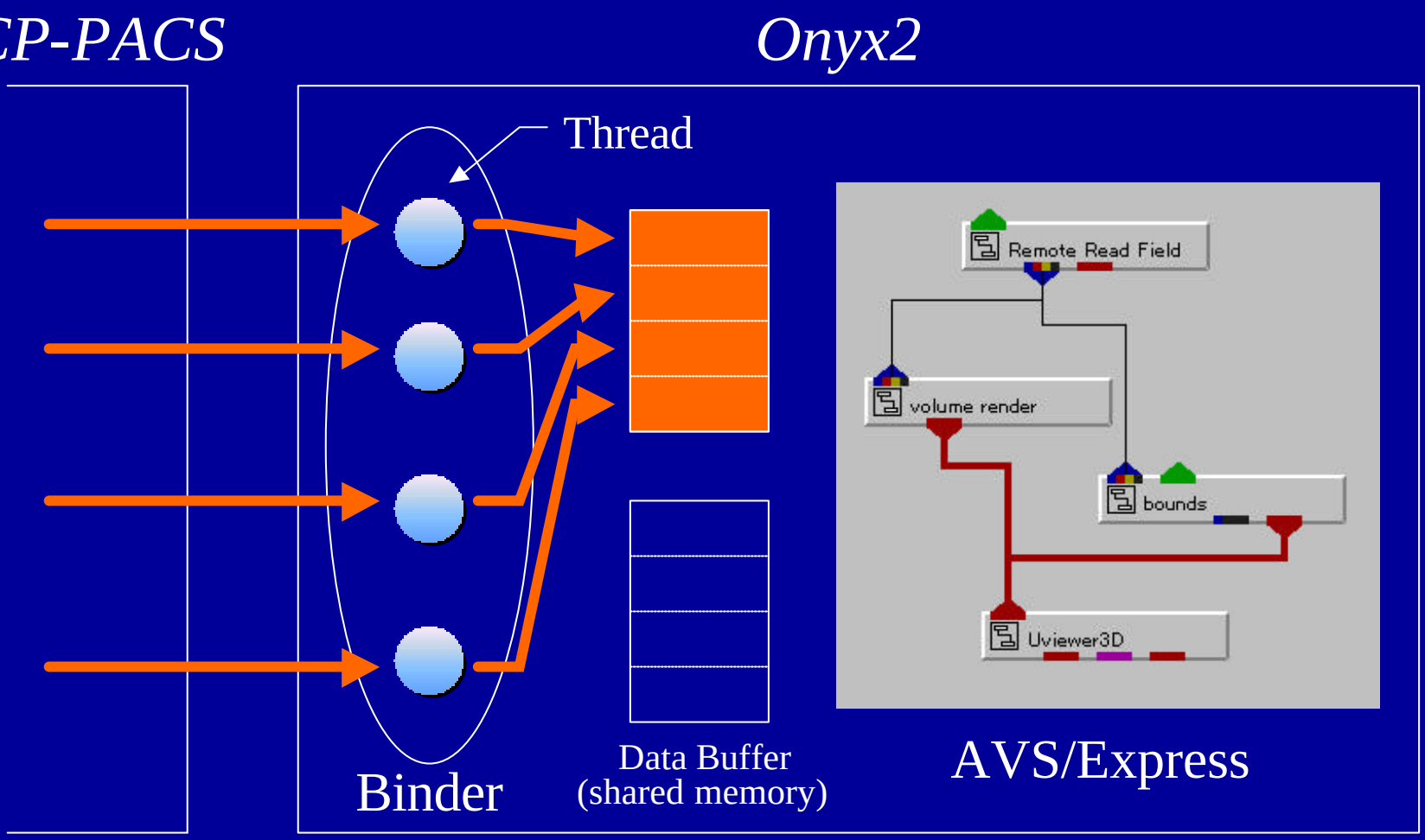
- 並列入出力システム（PIO）
  - アプリケーションプログラムから直接外部のマシンへデータ転送
  - ユーザはチャネルの本数を意識する必要がない
  - 静的負荷分散だけではなく動的負荷分散も行なう

並列入出力システムとAVS/Express の接続

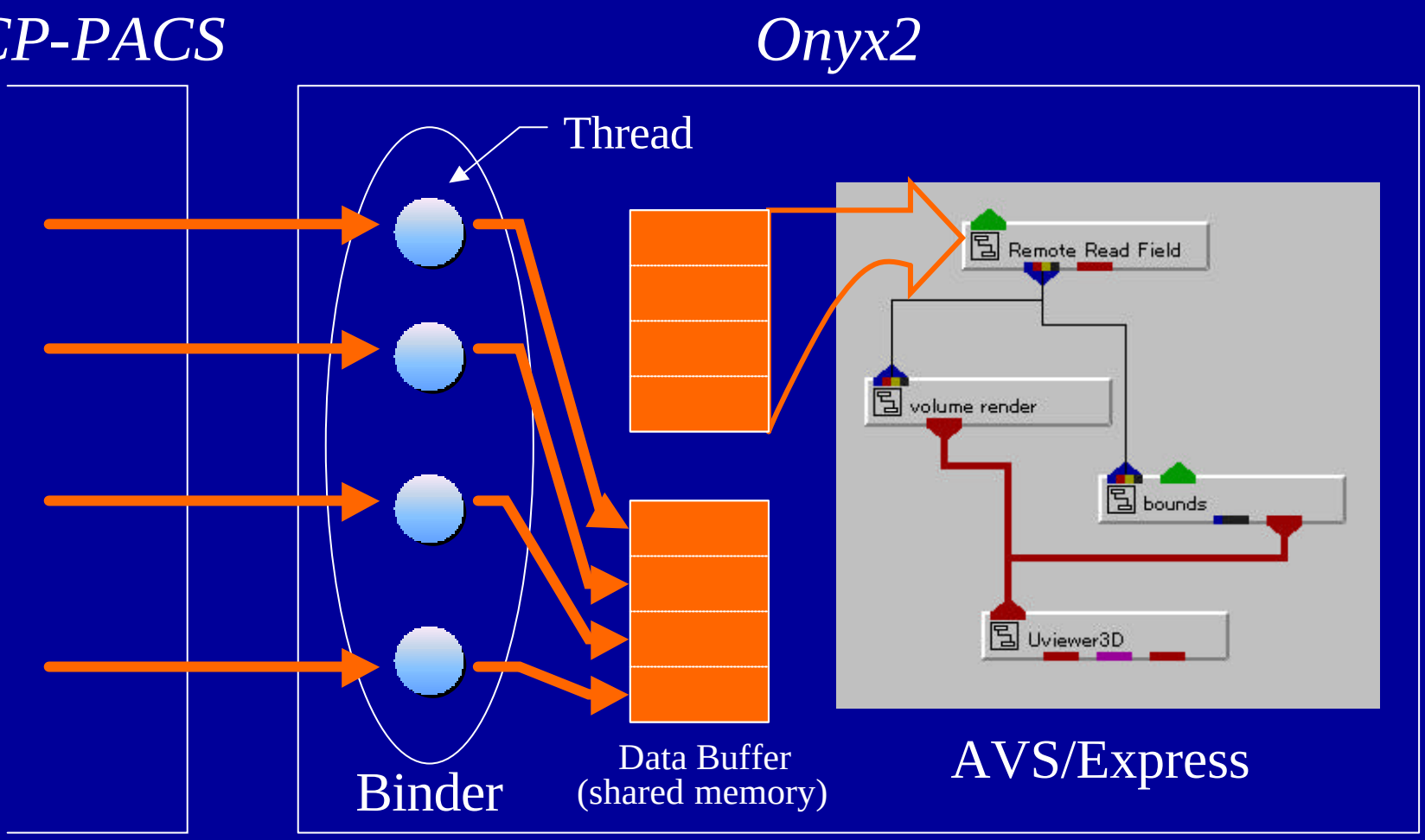
- AVS/Express 標準のデータ入力モジュールへのデータ供給はファイルからしか行なえない
- AVS/Express は逐次動作するので、データ入力モジュールと他の処理の機能分散ができない

- 並列データ入力モジュール（リードモジュール）を実装
- リードモジュールはデータを AVS/Express の内部データ形式に変換するだけ
- 並列チャネルからのデータ入力部分は別プロセス（Binder)として実装
- Binder とリードモジュールは共有メモリを介して、データの受け渡しをする

AVS/Expressへのデータ入力



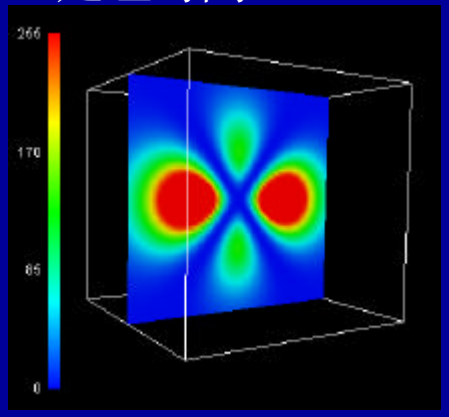
AVS/Expressへのデータ入力



可視化システムのボトルネック

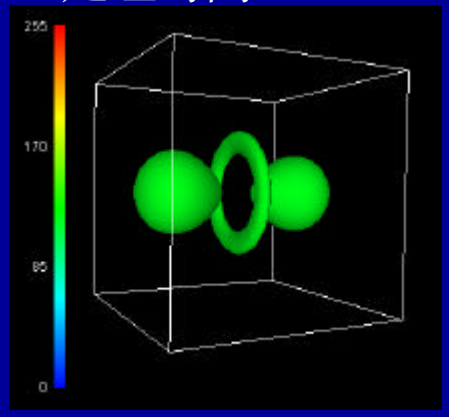
- AVS/Express内部の処理はすべて逐次処理
- Onyx2のグラフィックハードウェアを適用できないような可視化の場合、そこがボトルネックになる
- グラフィックハードウェアが適用できないような可視化
  - *ボリュームレンダリング*

処理時間：0.1sec



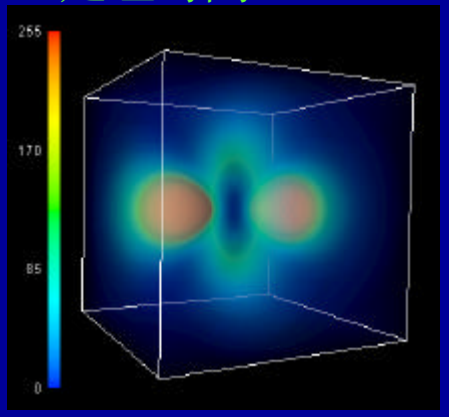
contour

処理時間：0.1sec



isosurface

処理時間：30sec



volume rendering

ボリュームレンダリングの高速化

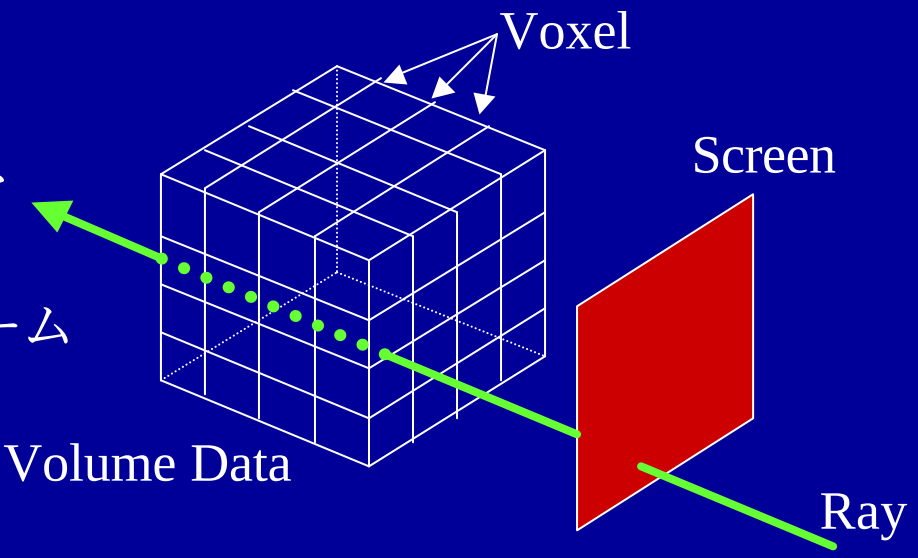
- 並列ボリュームレンダリングの実装
- Onyx2 or CP-PACS ?

	Onyx2への実装	CP-PACSへの実装
速度		◎ 高並列実行環境
操作性	◎ AVS/Express を利用可能	
実装の容易さ	◎	

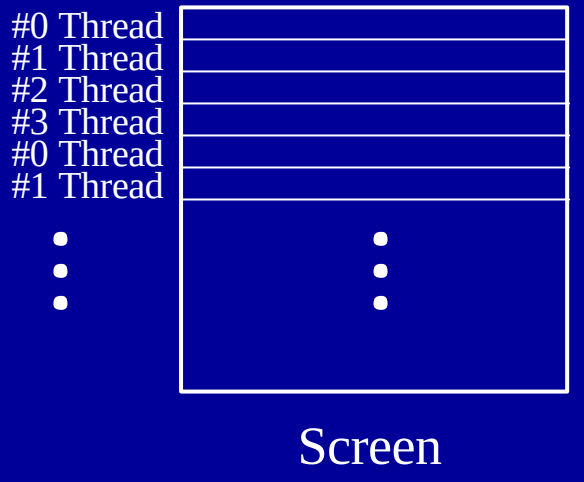
Onyx2への実装を採用
AVS/Expressの枠組みを崩さずに
並列ボリュームレンダリング･モジュールを実装する
（AVS/Expressの1モジュールとして動作するようにする）

アルゴリズム

- レイキャスティング
  - 視線方向に重なったデータ値を混合し、スクリーンに投影する
  - 不透明度を与えることでボリューム全体を半透明表示する



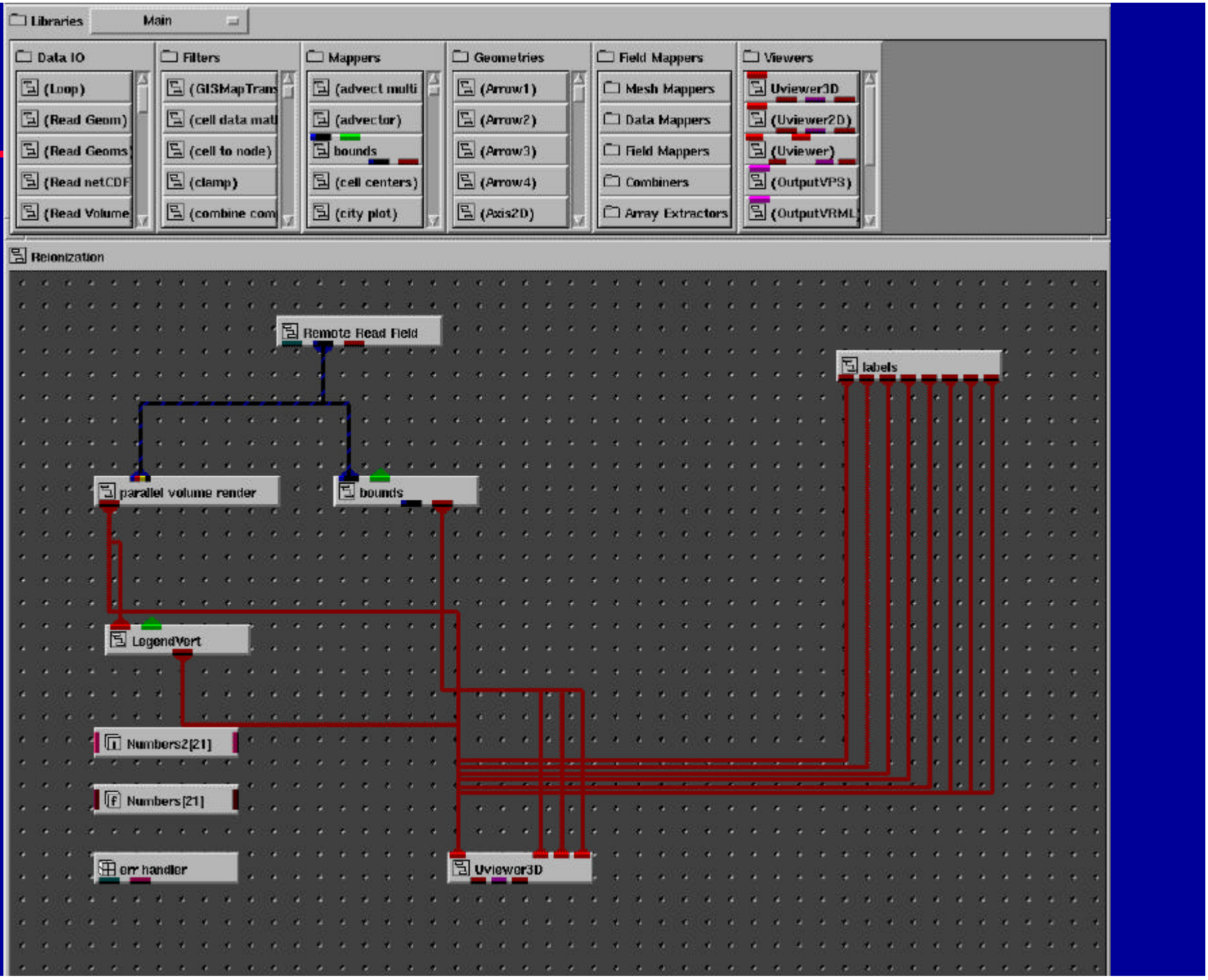
- スクリーンをスキャンライン単位で分割しそれをスレッドにサイクリックに割り当てる
  - 各スレッドの処理は独立
  - ボクセルデータに対して読み出し処理しか起こらない → スレッド同士の同期による待ち時間は生じない



AVS/Expressへの実装

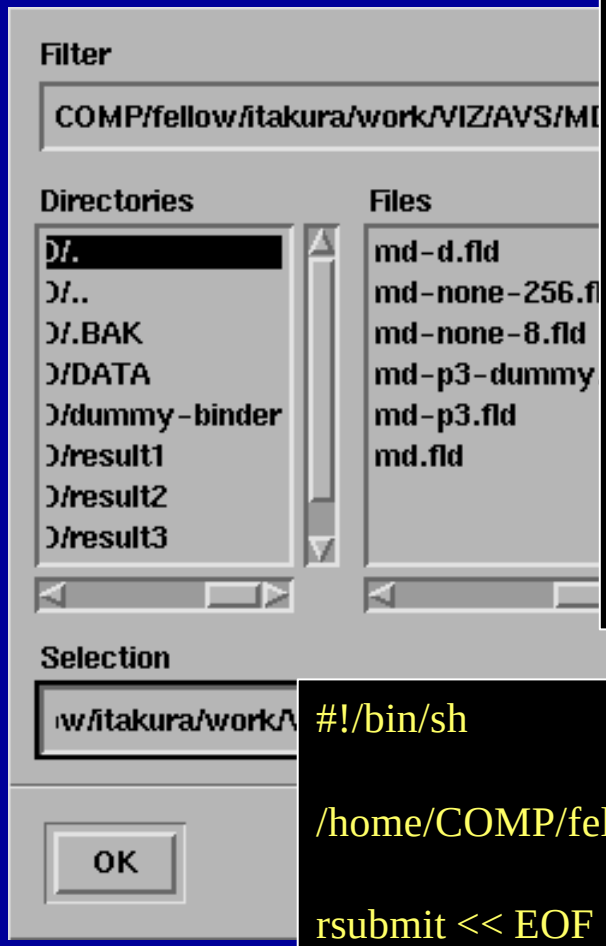
- 前処理に時間をかけないアルゴリズム
  - 一般的な入力データ→ スタティックデータ
  - *本システムが対象とするデータ→ ダイナミックデータ*
  - 入力データが動的に変化するため、あらかじめ高速レンダリング用データを作り、それを再利用する手法が使えない
- 高速化手法
  - Spatial Data Structure (8分木) による透明なボクセルのスキップ
  - Shading Lookup Table による輝度値計算の簡略化
- C言語 ＋ pthread で実装

標準モジュールと同様なレンダリング
標準モジュールと置き換え可能）


Libraries
Main
Data IO
(Loop)
(Read Geom)
(Read Geoms)
(Read netCDF)
(Read Volume)
Filters
(GISMapTrans
(cell data mat
(cell to node)
(clamp)
(combine com
Mappers
(advect multi
(advector)
bounds
(cell centers)
(city plot)
Geometries
(Arrow1)
(Arrow2)
(Arrow3)
(Arrow4)
(Axis2D)
Field Mappers
Mesh Mappers
Data Mappers
Field Mappers
Combiners
Array Extractors
Viewers
Uviewer3D
(Uviewer2D)
(Uviewer)
(OutputVPS)
(OutputVRML)
Relonization
Remote Read Field
labels
parallel volume render
bounds
LegendVert
Numbers2[21]
Numbers[21]
err handler
Uviewer3D

コントロールパネル（GUI）から入力データの情報を与える。

AVS/Expressの標準のRead Moduleと互換性を持たせる。




```
# AVS field file
#
ndim = 3
dim1 = 16
dim2 = 16
dim3 = 16
nspace = 3
veclen = 1
data = byte
field = uniform
receiver = /home/COMP/fellow/itakura/work/VIZ/AVS/MD/md-p3.sh
```

```
#!/bin/sh

/home/COMP/fellow/itakura/pio/run-20000630/piosrc/MD/receiver pilot3 &

rsubmit << EOF
cd /home/COMP/fellow/itakura/nfs/ccphfs/PIO/MD/run_script/pilot3
qsub go
exit
EOF
```

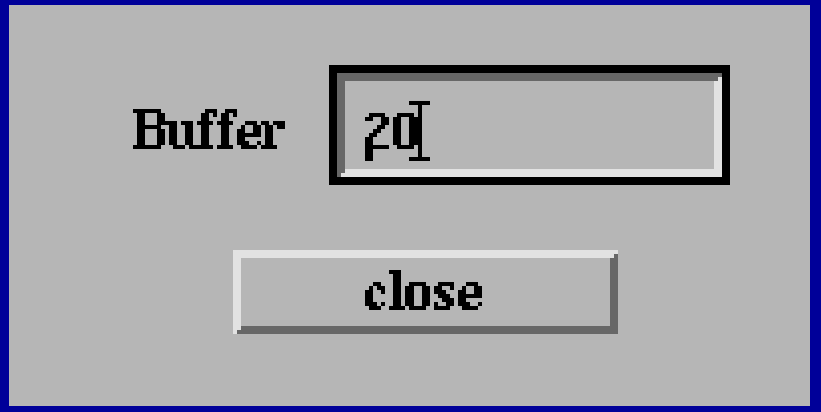
計算サーバからデータが送られてくると、描画可能となる。

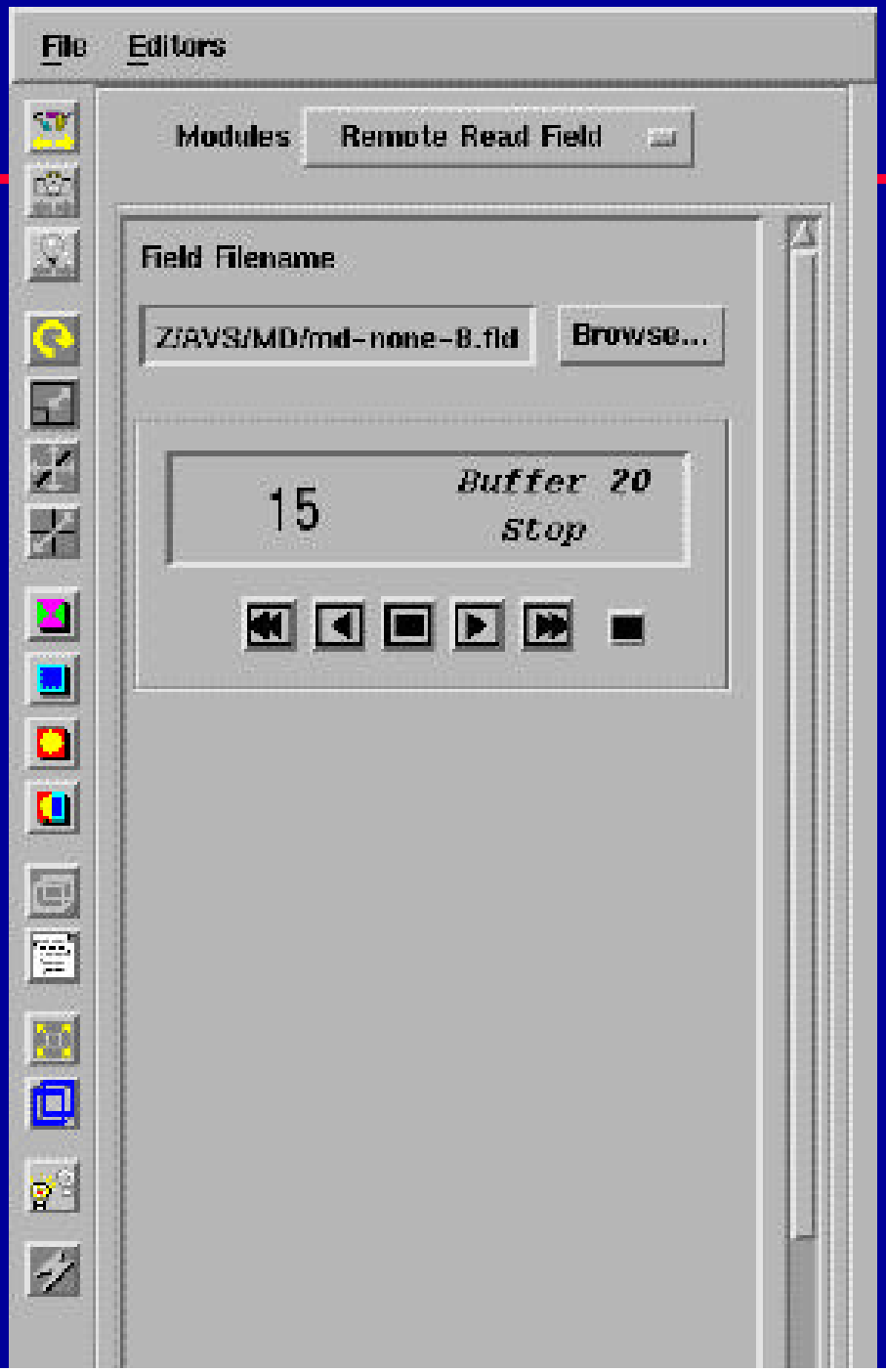
停止、1面分の描画、連続描画のコントロールが可能

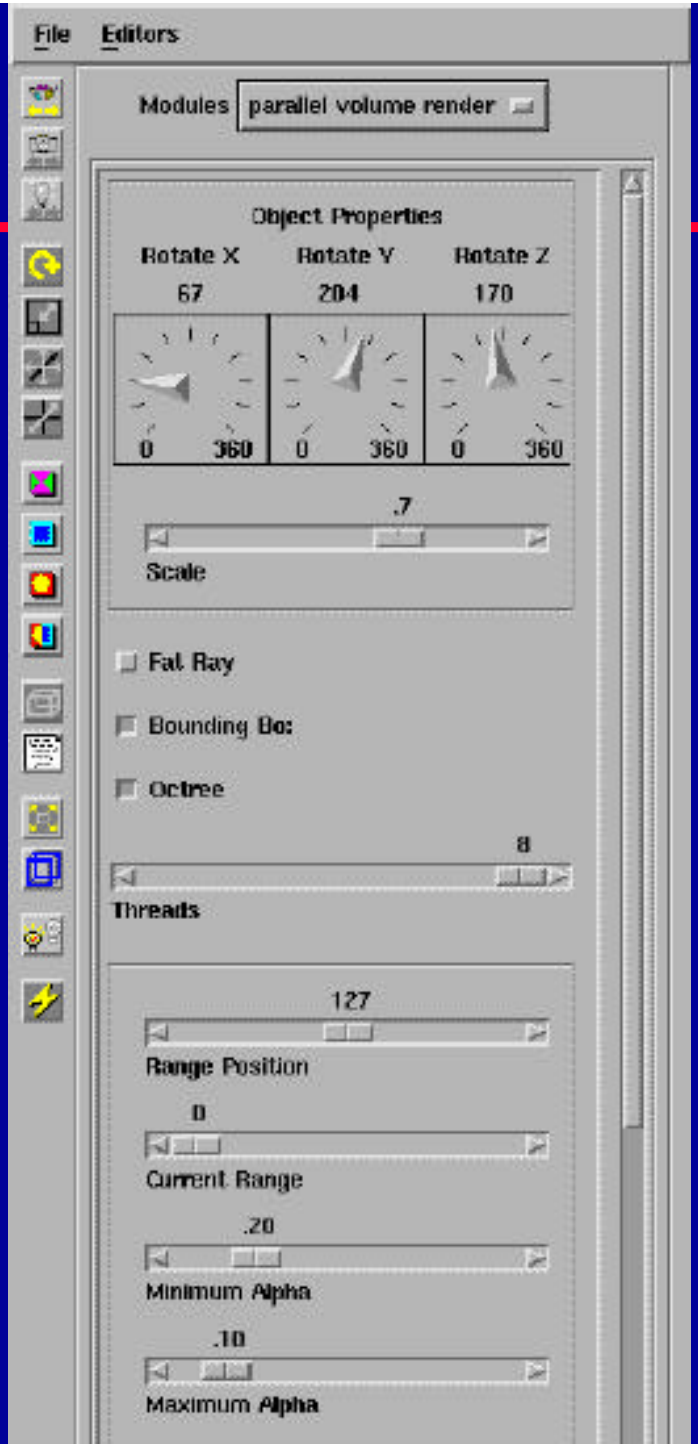
Remote Read Field にバッファリング機能を付加している。

あらかじめ用意したバッファに raw data を保持しておくことで、

巻き戻して別の角度から描画することもできる。

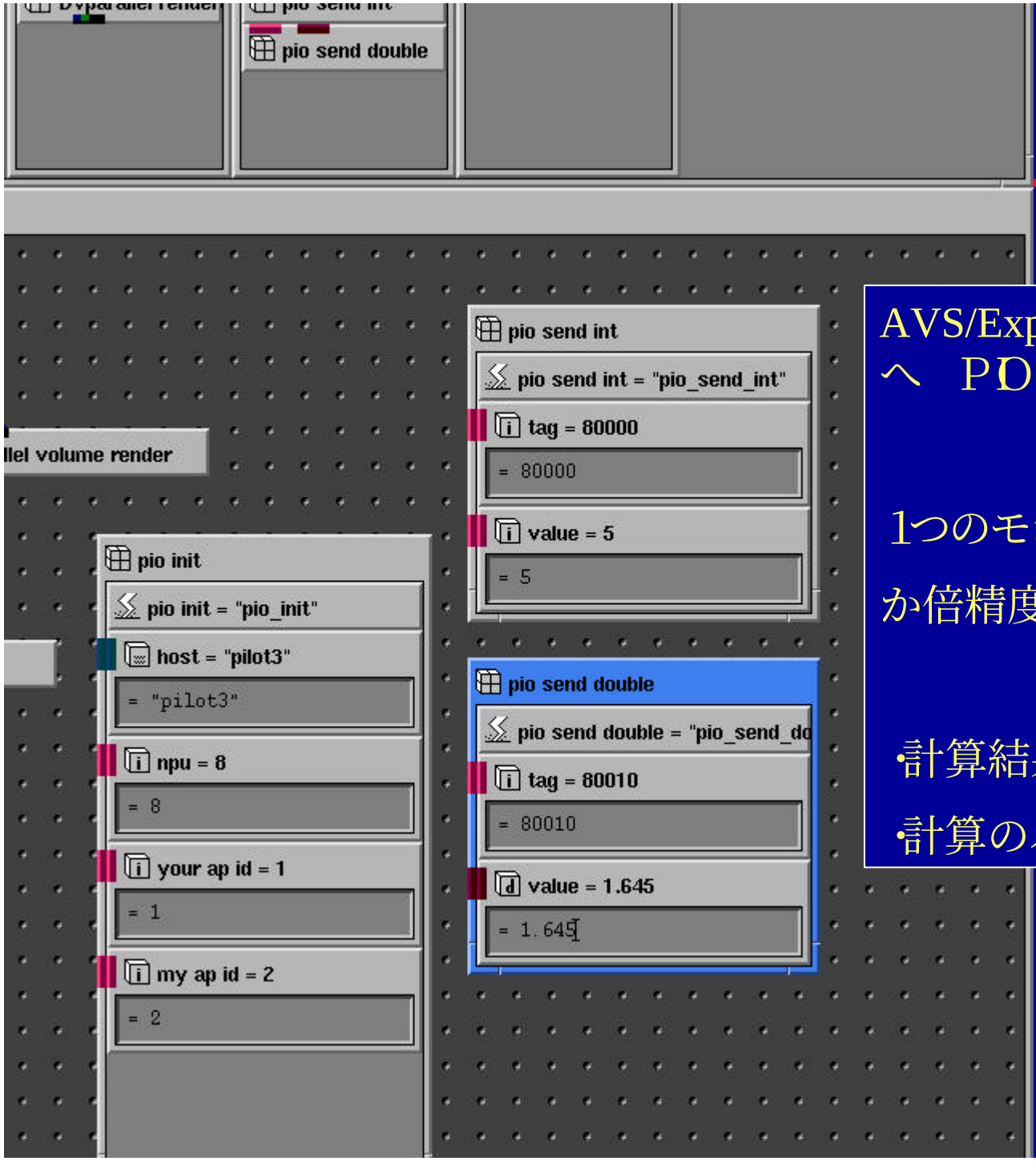



File
Editors
Modules
Remote Read Field
Field Filename
Z:/AVS/MD/md-none-8.fld
Browse...
15
Buffer 20
Stop



並列VRエンジンに対する操作：

1. 物体をX, Y, Z軸での回転
2. 物体を0.1 から1.0 倍まで拡大縮小
   これとは別に、ビューアーで拡大縮小が可能
3. Rayの本数を1/4にしてレンダリングを高速化する。
4. Bounding Boxの表示
5. Octreeによる最適化
6. 任意のスレッド数による実行
7. 不透明度の設定は 2直線による指定



AVS/Expressから計算プログラムへ　PIOによるデータ転送を行う。

1つのモジュールから整数値か倍精度値のスカラーを送る。

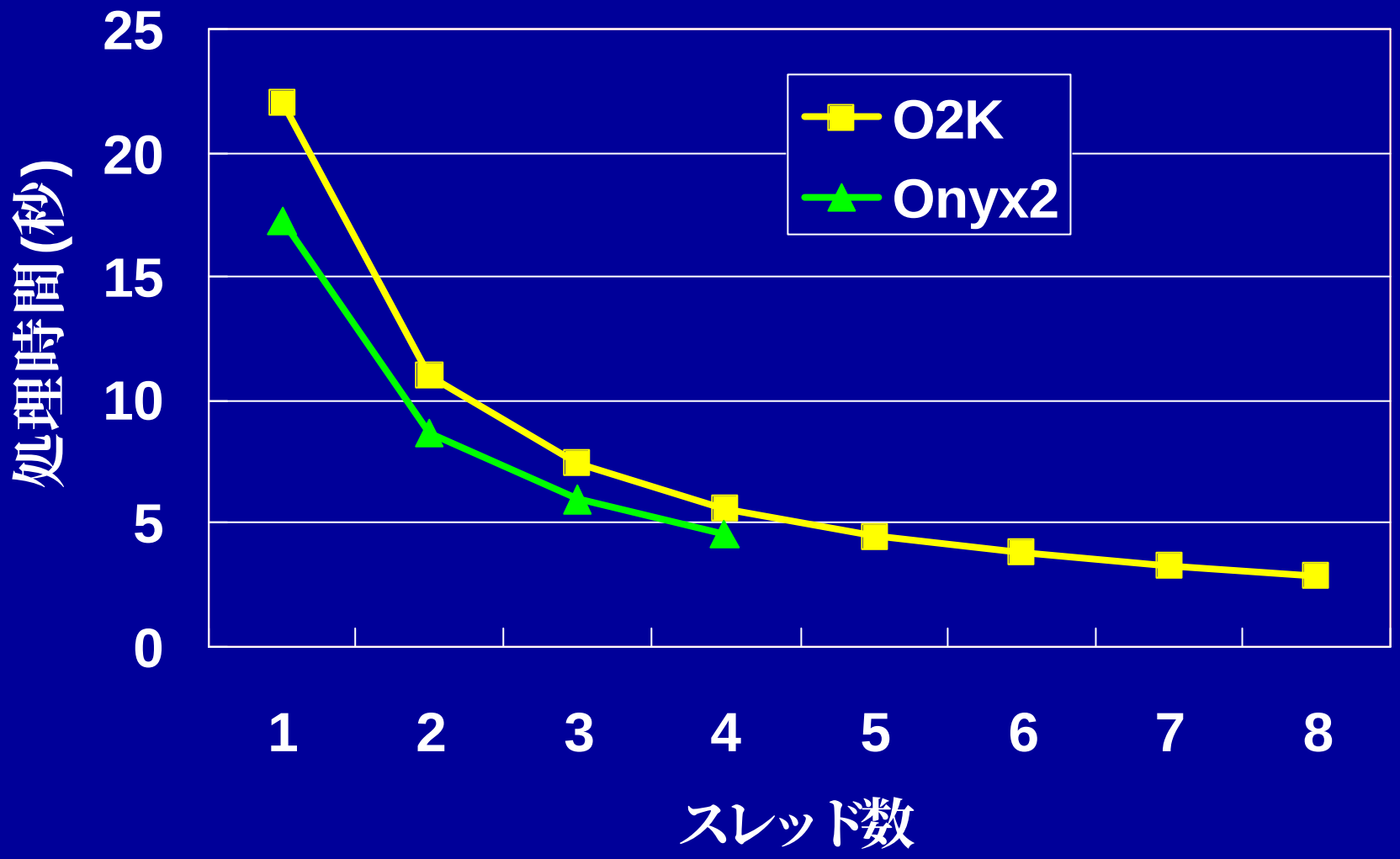
- 計算結果の出力をコントロール
- 計算のパラメータの変更

並列ボリュームレンダリングエンジンの性能評価

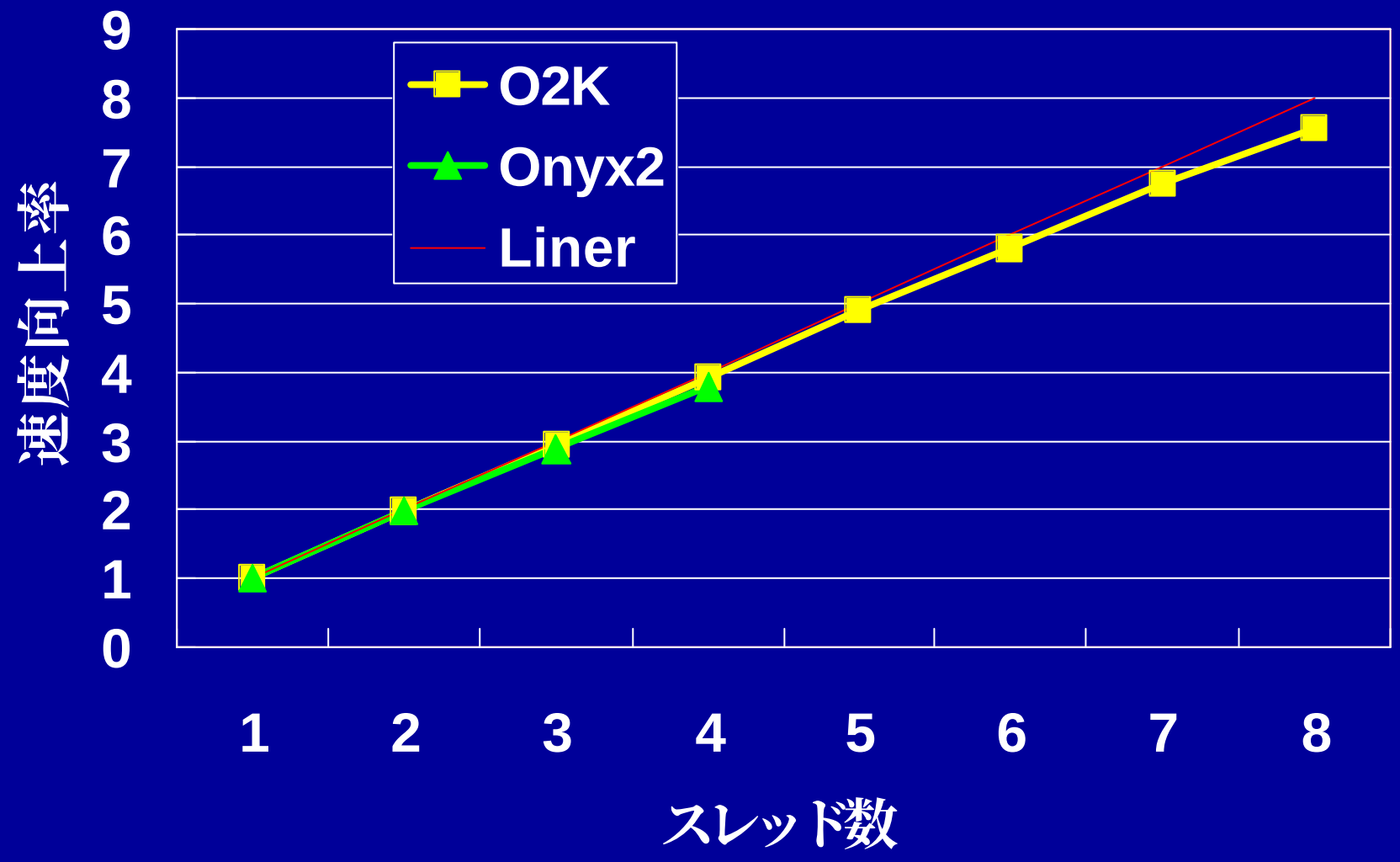
- 評価方法
  - 3次元ボリュームレンダリングデータ :あらかじめメモリに読込
  - 画像を2次元データ配列に格納するまでの時間

- 条件
  - 宇宙初期の水素ガスの電離状態を表すデータ
    （128×128×128 ボクセル）
  - 画像サイズ：512×512 ピクセル
  - 可視化サーバ：Onyx2　(4CPU, 250MHz)
    Origin2000 (8CPU, 195MHz)

Rendering Time (Library)



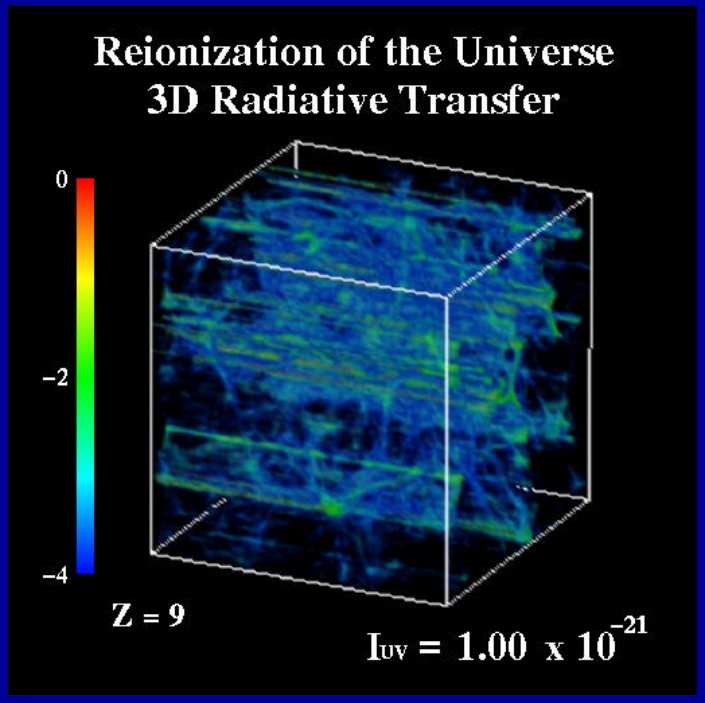
Rendering Time (Library)



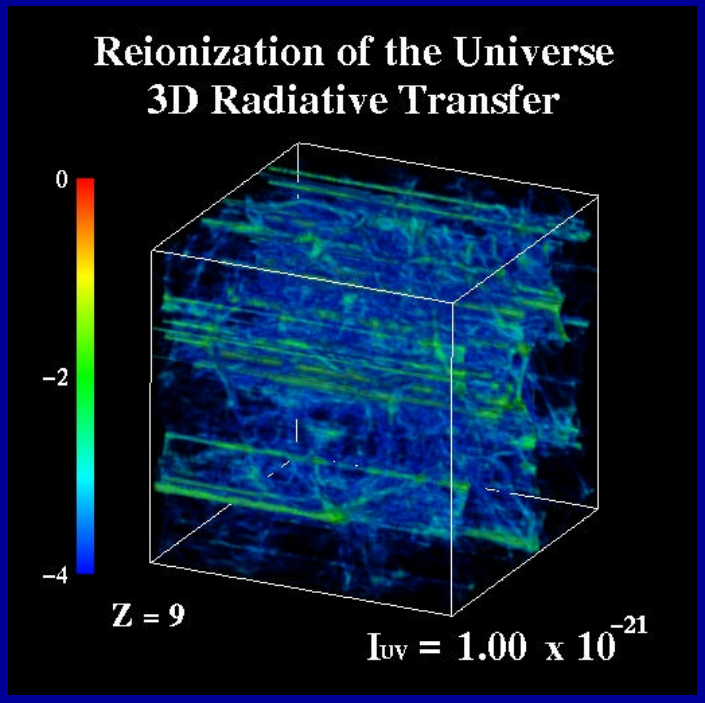
AVS/Expressモジュール化した性能評価

- AVS/Express標準のボリュームレンダリング・モジュールと並列ボリュームレンダリング・モジュールの比較をする
- 評価方法
  - 時系列データをファイルから読み込みながら、連続でボリュームレンダリングを行ない、その全実行時間を測定する
- 条件
  - どちらのモジュールでも同じような画像になるようにパラメータを調整
  - 宇宙初期の再電離過程の3次元輻射輸送シミュレーションデータ（128×128×128 ボクセル × **60 ステップ**）
  - 画像サイズ：512×512 ピクセル
  - 可視化サーバ：Onyx2 (4CPU, 250MHz)
    Origin2000 (8CPU, 195MHz)

レンダリング画像

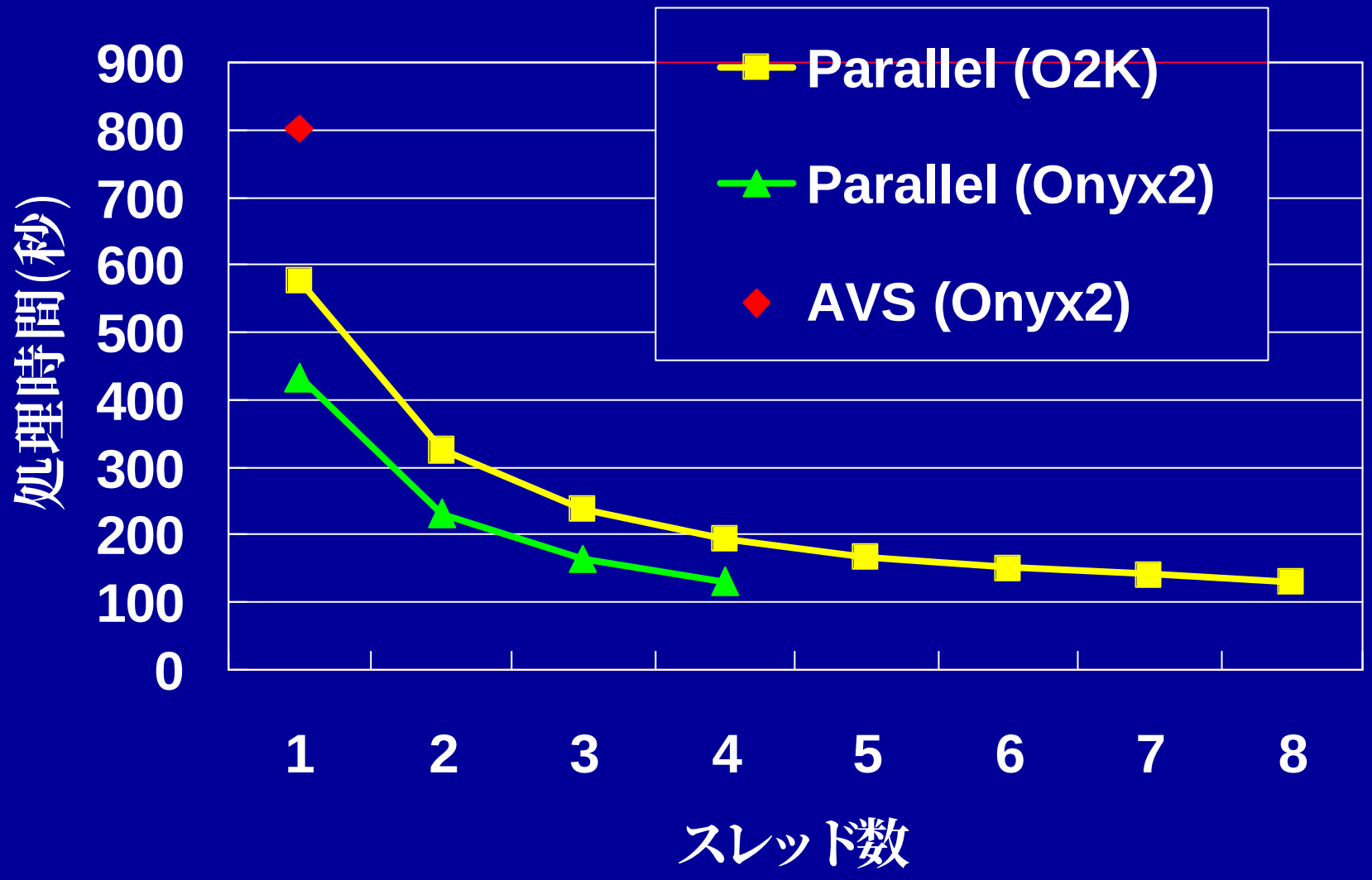


並列ボリュームレンダリングによる画像

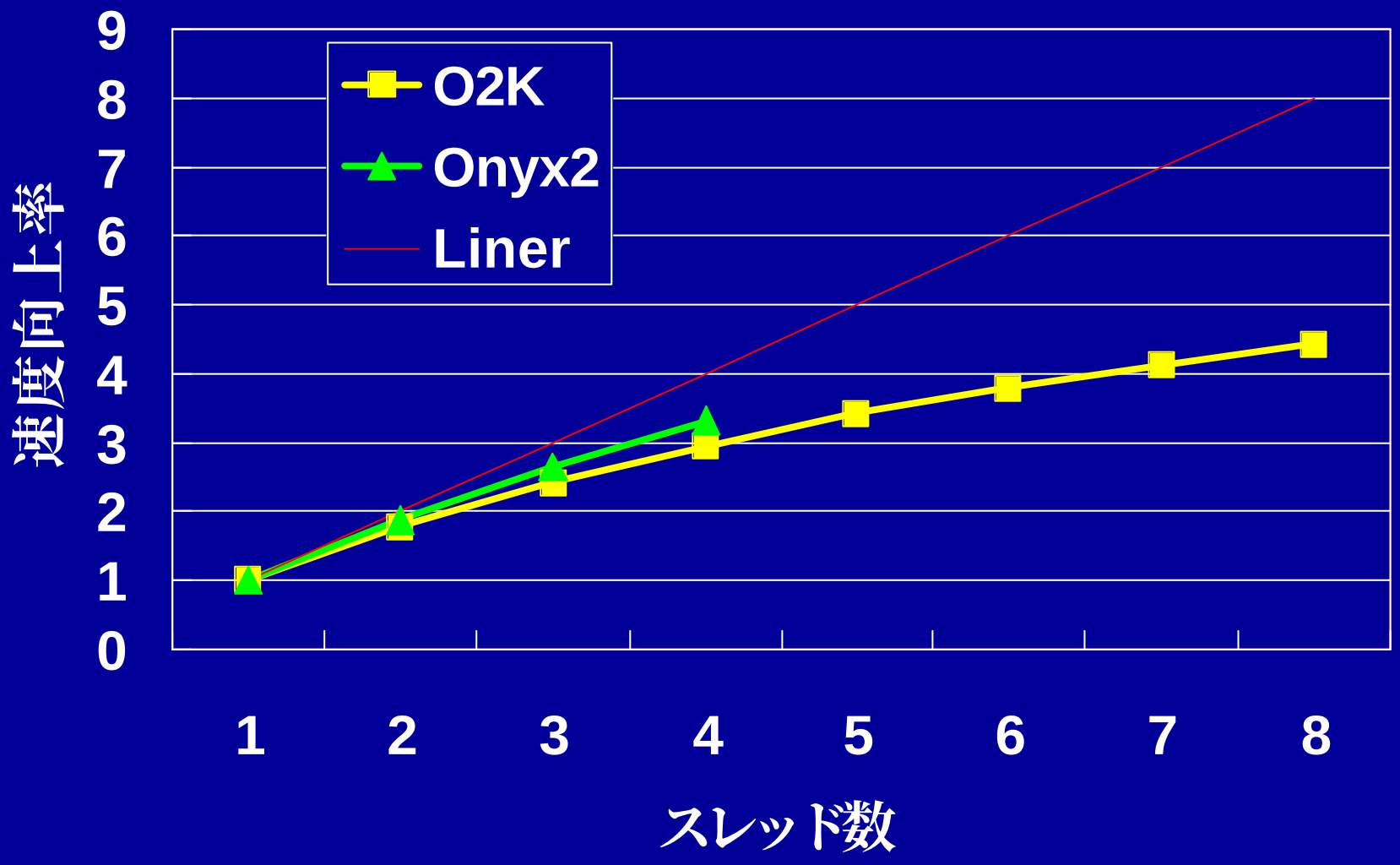


標準モジュールによる画像

Rendering Time (AVS module)



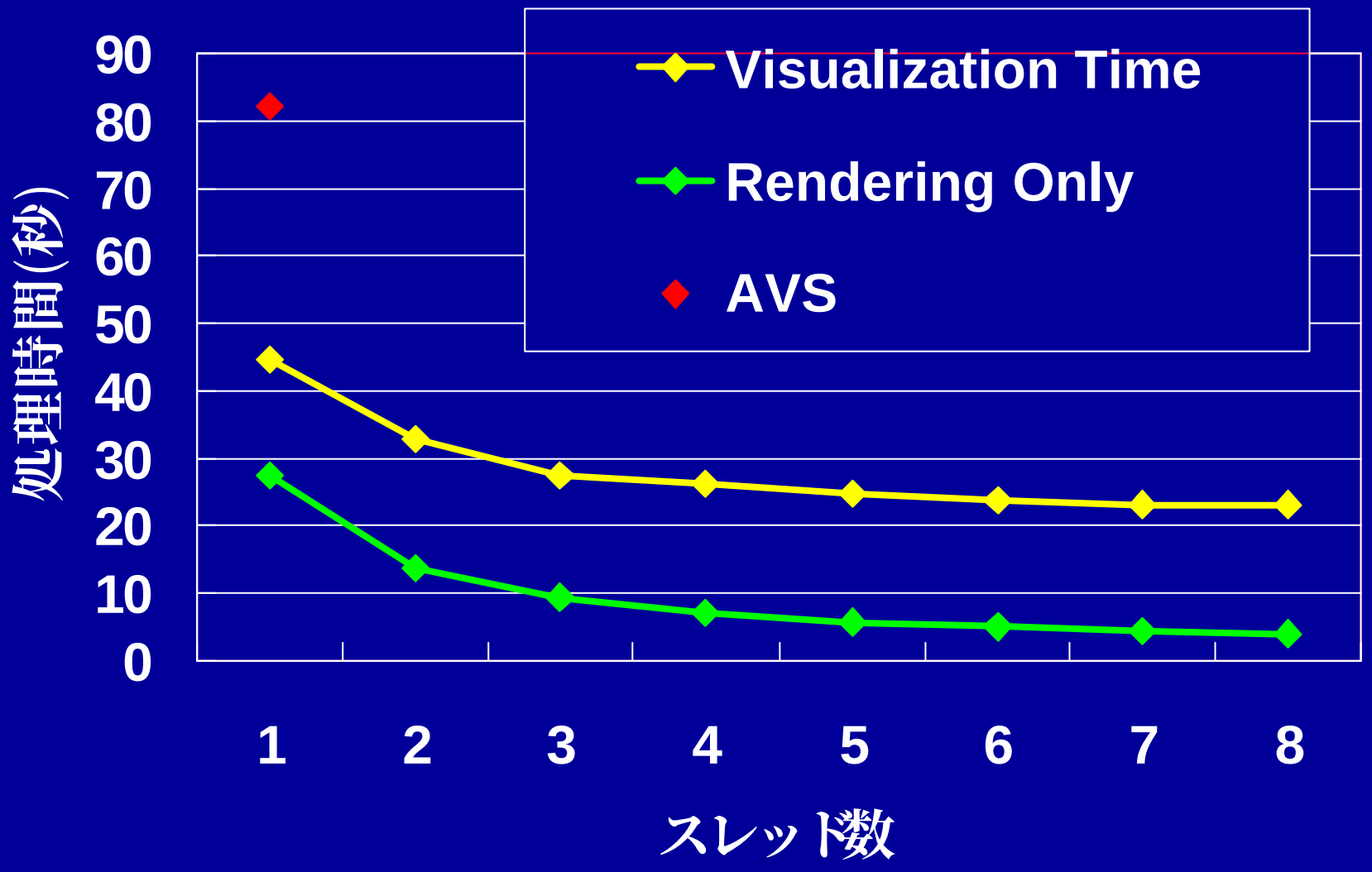
Rendering Time (AVS module)



PIOとの連動協調動作の性能評価

- 評価方法
  - CP-PACSの256PUを用いて数値計算を行う。
  - 可視化サーバ側はbinderプログラムとAVS/Express を同時に実行する。
  - 連続でボリュームレンダリングを行ない、その全実行時間を測定する
- 条件
  - 分子動力学シミュレーション (16K個の液体アルゴン分子)
    (16×16×16 ボクセル × 20 ステップ)
  - 画像サイズ：512×512 ピクセル
  - 可視化サーバ：Origin2000 (8CPU, 195MHz)

Rendering Time (AVS module + PIO)

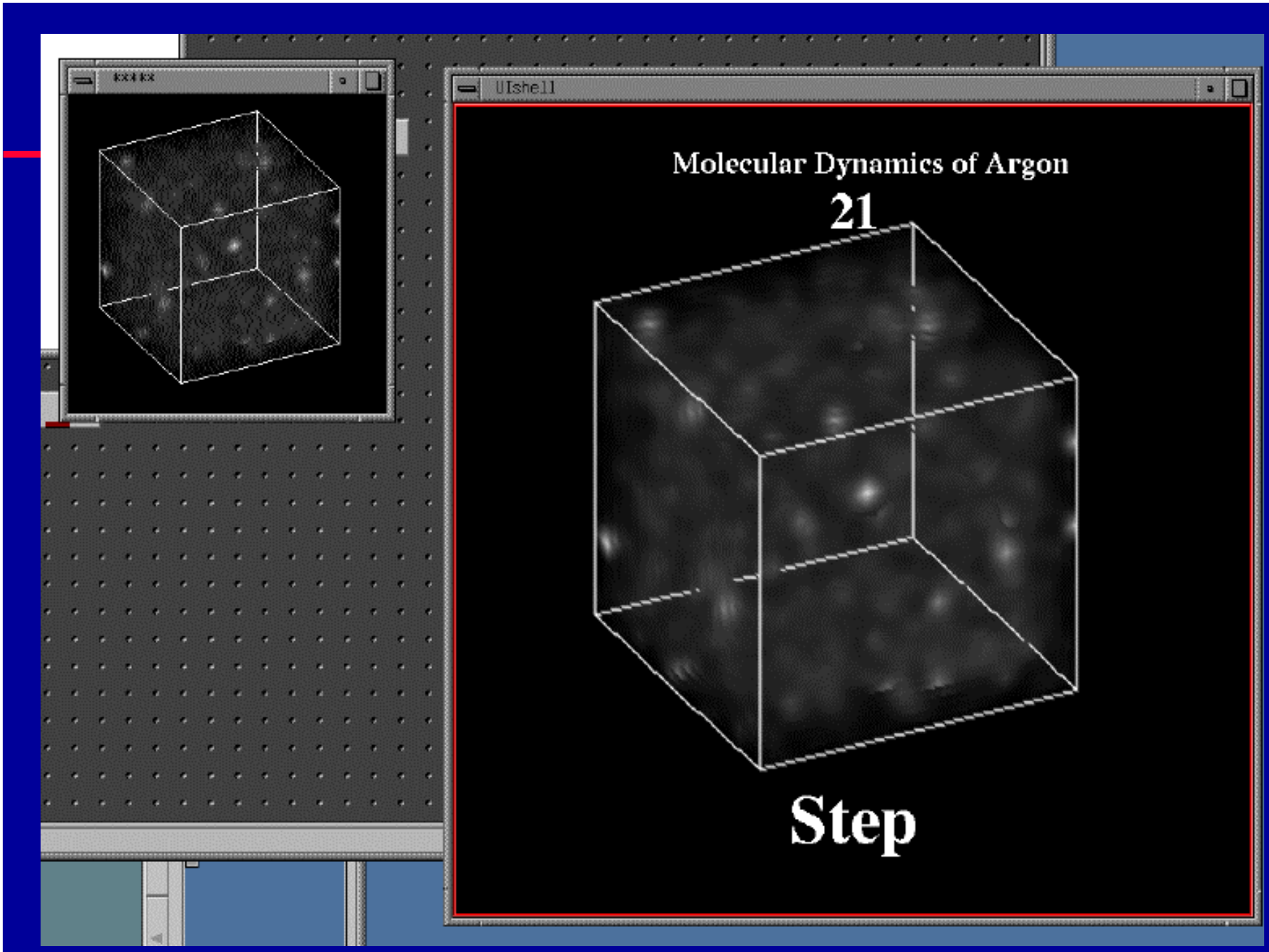


Rendering Time (AVS module + PIO)

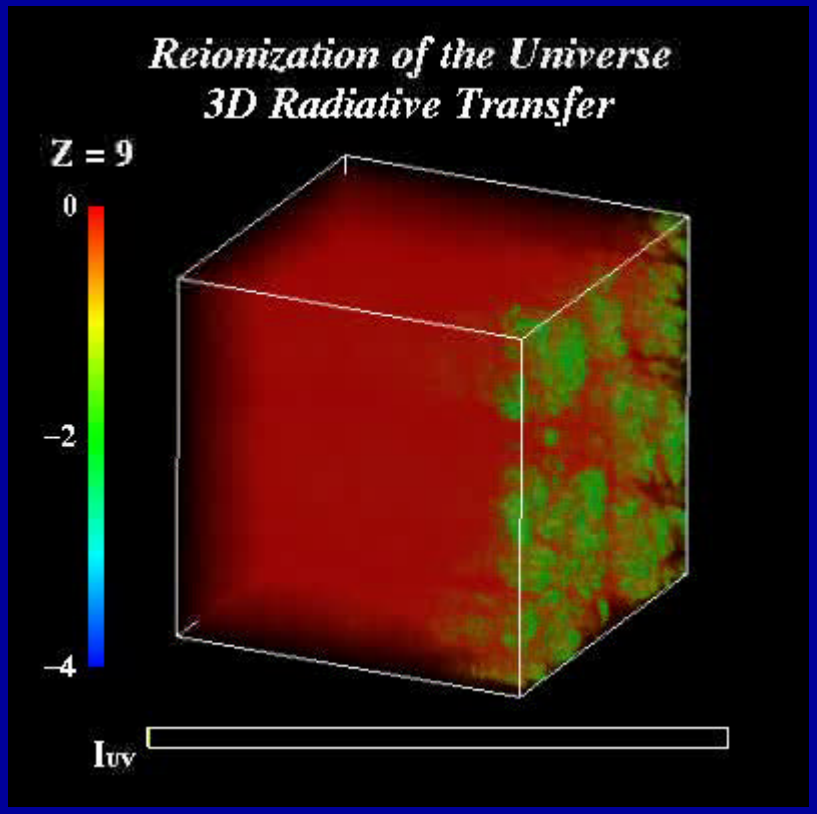
	AVS	並列レンダリングモジュール (スレッド数)							
		1	2	3	4	5	6	7	8
データ読込	0.8	0.7	0.8	0.8	0.8	0.8	0.8	0.8	1.1
レンダリング		27.6	13.9	9.4	7.2	5.8	4.9	4.3	4.0
その他		16.5	18.3	17.3	18.1	18.1	18.1	18.1	17.9
可視化合計	81.6	44.1	32.2	26.7	25.3	24.0	23.0	22.4	21.9
総処理時間	82.4	44.8	33.0	27.5	26.0	24.8	23.8	23.2	23.0

[秒]

その他はレンダリング終了後の画面描画オーバヘッド
X11ライブラリを用いた単純な画像出力で9秒程度まで短縮できる


xxxxx
UIshell
Molecular Dynamics of Argon
21
Step

可視化例



Reionization of the Universe

CP-PACS 2048PU 128IOU

+

16 channels

+

Origin2000 8CPU

⇩

Real-Time Visualization

まとめ

- 並列可視化システムの実装
  - 数値計算と可視化を同時並行的に実行
  - 並列入出力システムを用いてデータ転送を並列化
  - 並列ボリュームレンダリングをAVS/Expressのモジュールとして実装
- 様々なプラットホームへ適用可能
  - PIO → TCP/IPを用いて実装
  - AVS/Express → 汎用可視化ソフトウェア

今年度の残作業

- レンダリングの視点、スクリーンの位置変更をGUIで行うために
  - AVS5のレンダリングモジュール tracer に対して AVS/Express のビューアー Uviewer からフィードバックさせる方法を KGT が検討している。
  - Parallel VR モジュールを tracer と同等の I/F にする。
- クオリティ向上のために、 tracer の pthread による並列化