AICS

RIKEN Advanced Institute for Computational Science

大規模な計算結果の可視化における課題とアプローチについて

小野 謙二

理化学研究所 計算科学研究機構

可視化技術研究チーム

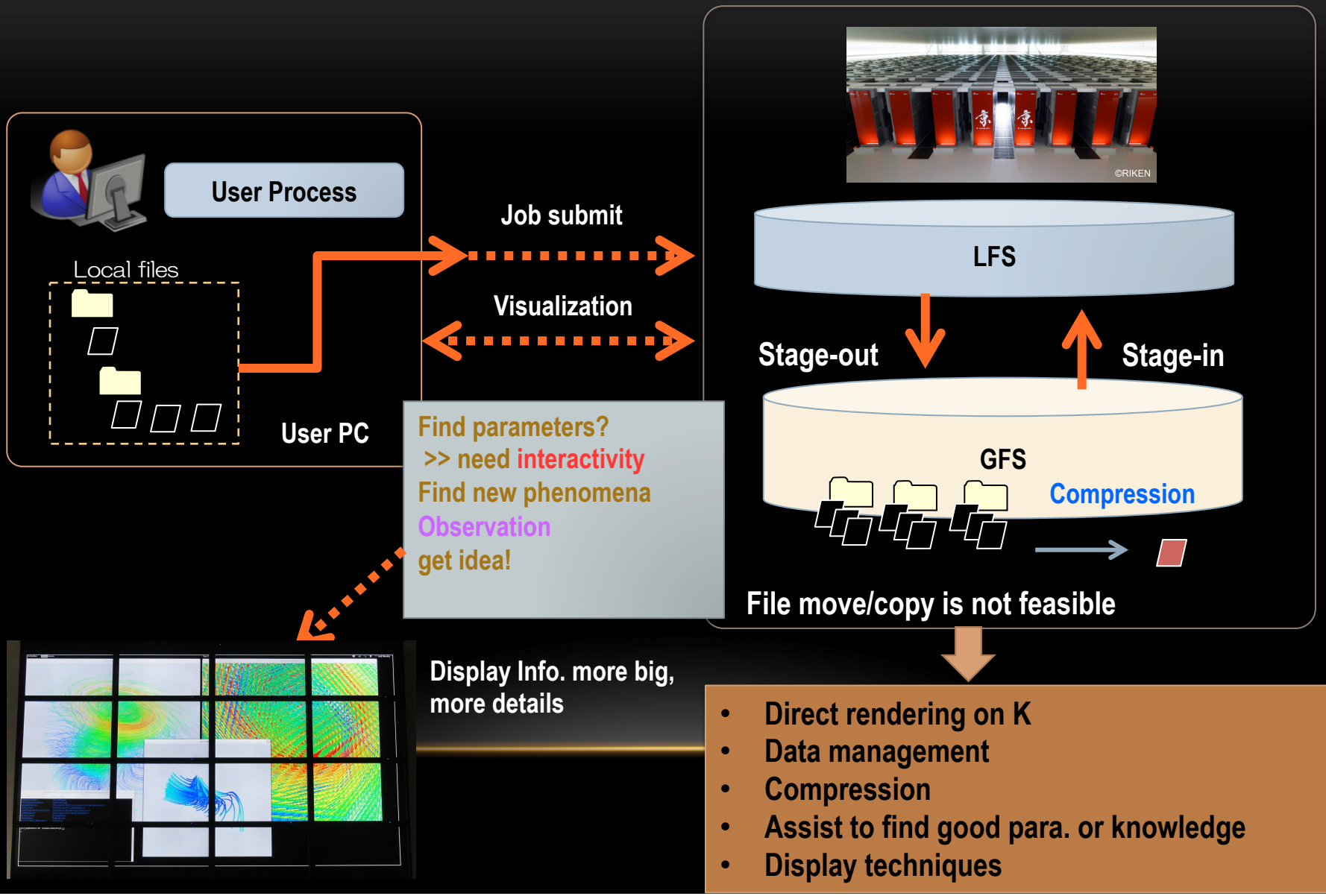
/ 神戸大学

CMSI Seminar 2013-12-2

TOC

- 大規模の課題
- 京における可視化
- どこに向かっていくか

BOTTLE NECK IN SIMULATION PROCESS



データが大規模になると困ること

- 扱いに困る
  - ファイルの数が多すぎて管理しきれない
  - 一つのディレクトリにたくさんのファイル > ls...
  - 手作業？
  - シミュレーションの次に後処理にデータを渡す
- 処理に時間がかかる
  - ファイルI/Oに時間がかかる
  - 並列化に向かない処理

1. ファイルハンドリング
2. ファイルI/O
3. データ圧縮
4. 大規模データ可視化

１．多数のファイルを管理する

- アプリケーション間，プロセス間にわたり，ファイルの受け渡しに利用する共通機能ライブラリを使う
- 抽象レベルでは同じアイデアが使えるが，実装はデータ構造毎に異なる
- まず，直交格子について，アプリケーション側の管理構造を策定
  - ファイル入出力性能は，サブモジュールとして考慮
  - 非構造データ，Octreeなどについても設計中

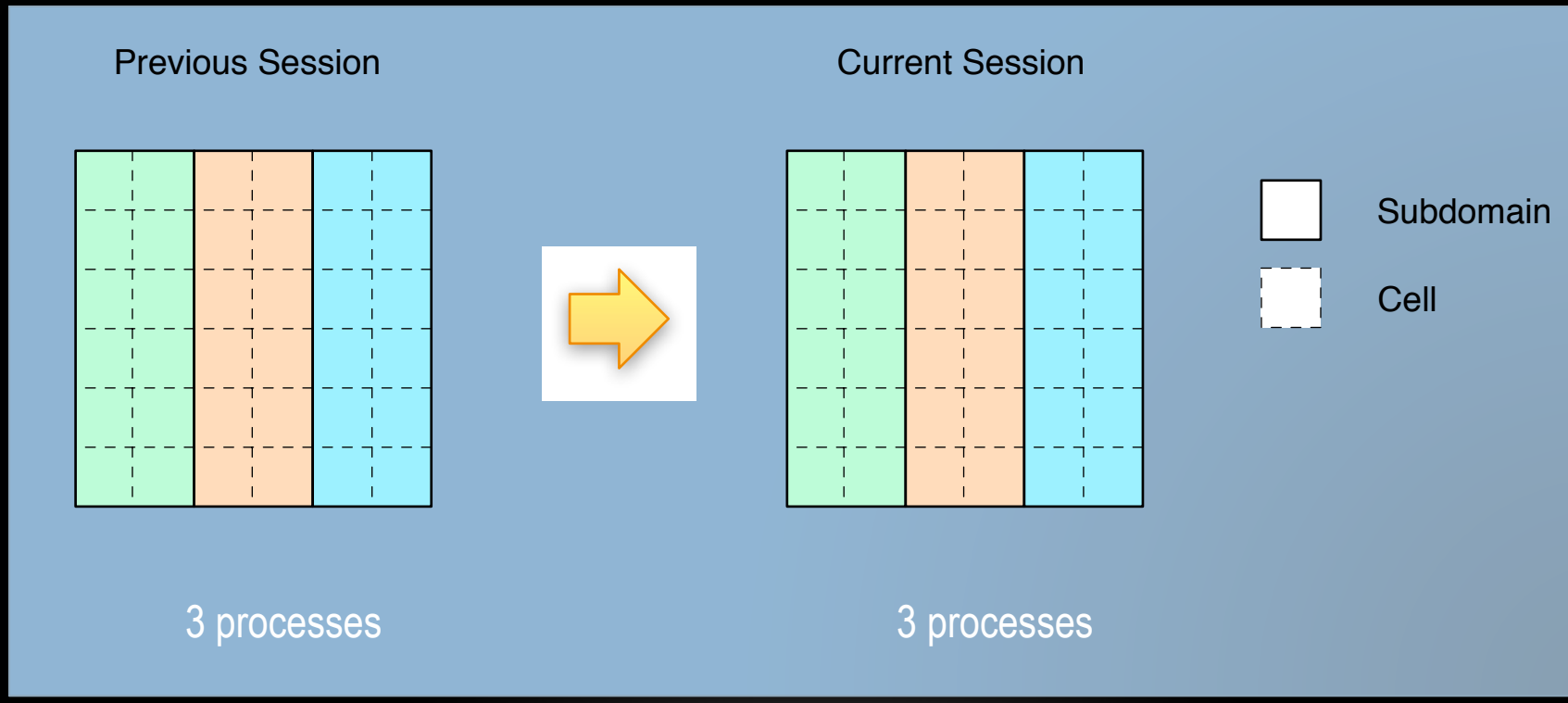
CIO LIBRARY

- 並列分散環境における直交格子データのファイル管理機能を提供

- 機能
  - 分散ファイルをメタデータにより管理（DFI file）
    - SPH, BOVファイルフォーマット
  - 様々なリスタートパターン
    - M x N再分配
    - 粗格子から密格子への内挿（格子幅は1/2）
    - ステージングへの対応
  - フォーマット変換
    - SPH, BOVフォーマットを軸として，PLOT3D, AVS, VTK形式へ変換

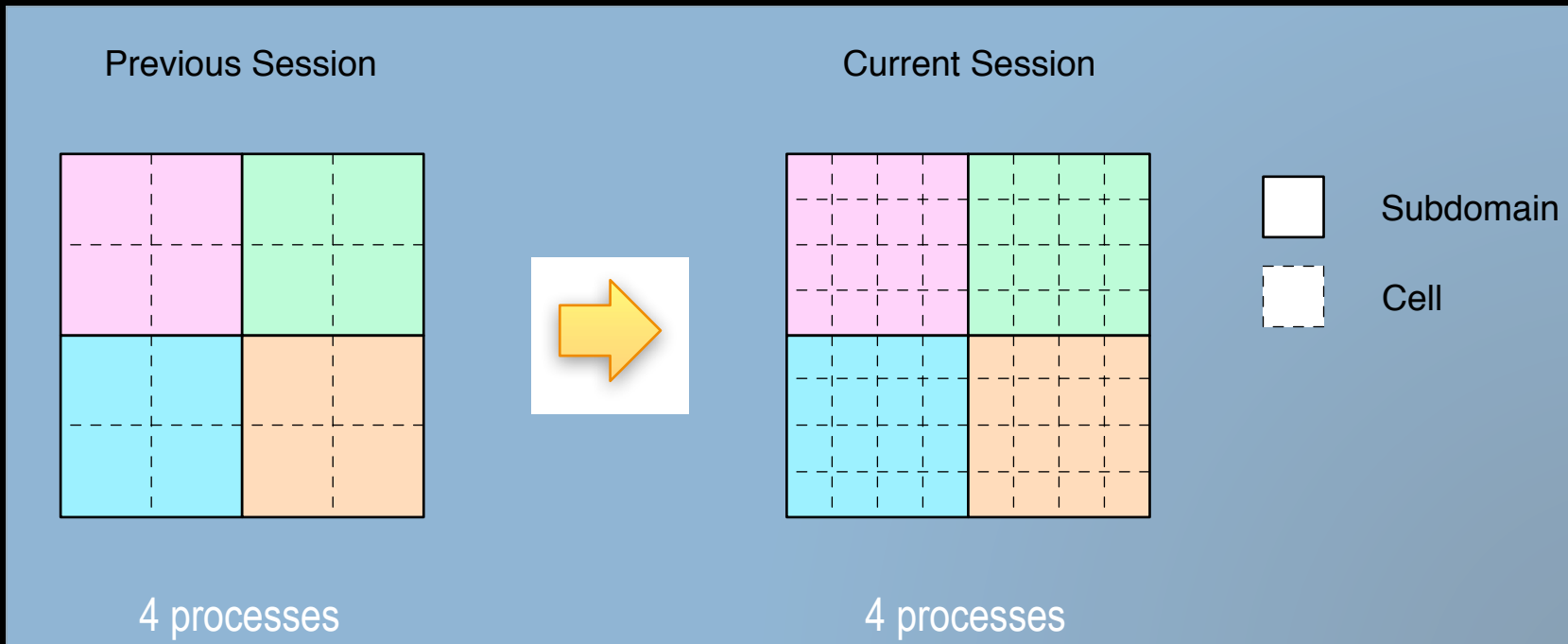
USE CASES 1

Same # of processes, same resolution => Std. restart



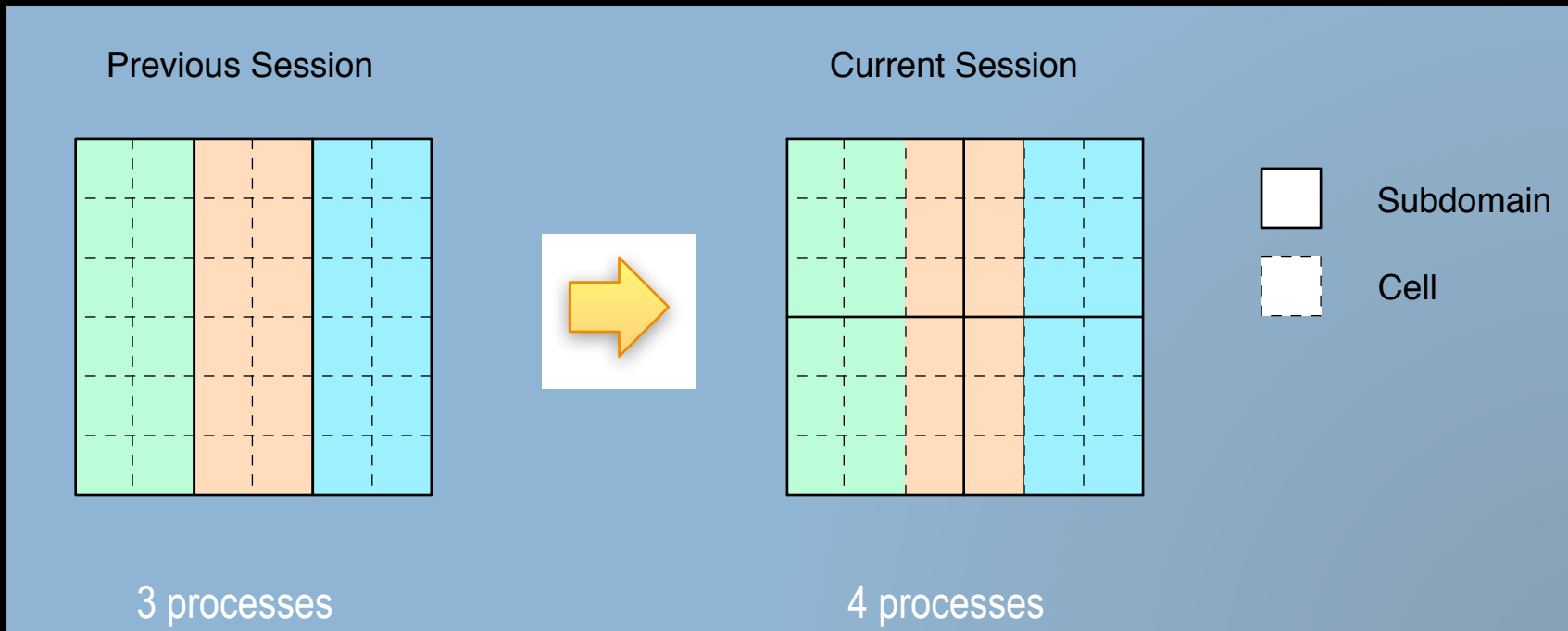
USE CASES 2

Same # of processes, different resolution => Refinement restart



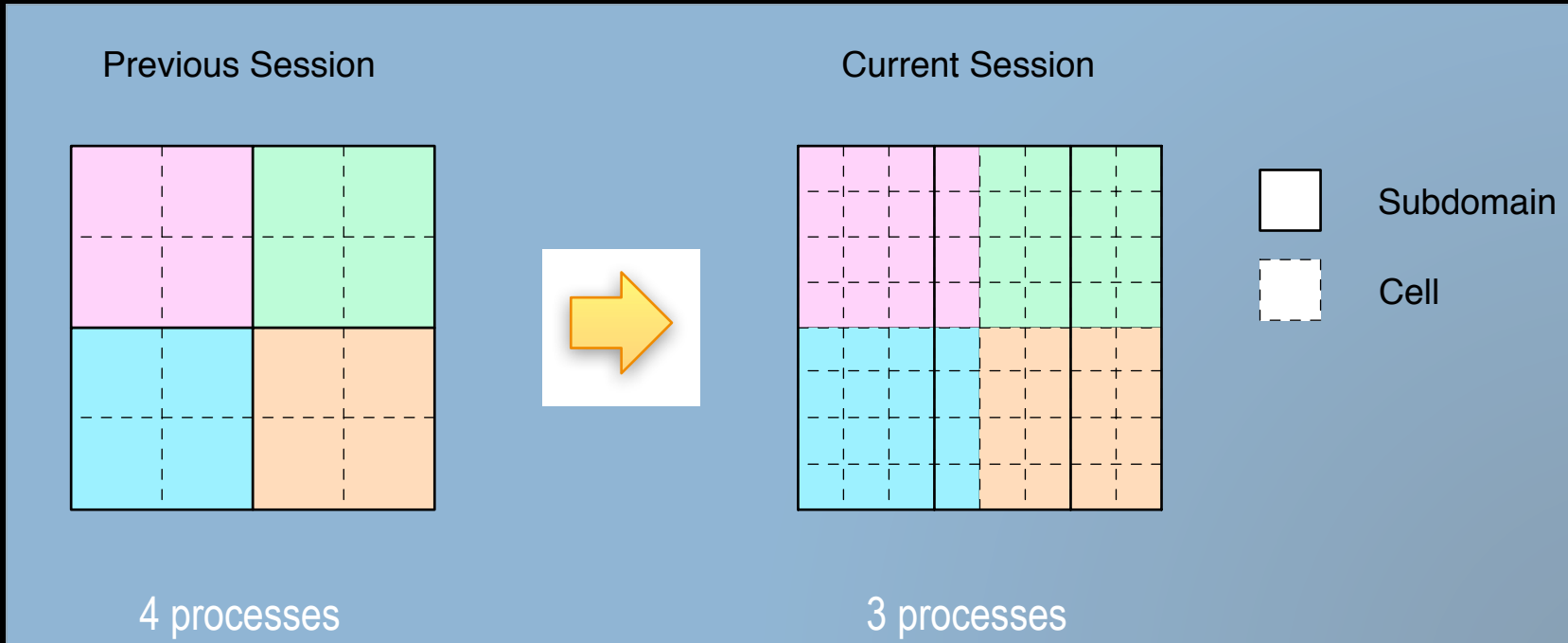
USE CASES 3

Different # of processes, same resolution => M X N restart



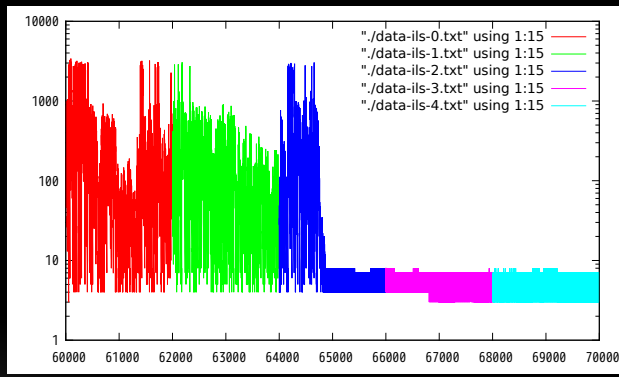
USE CASES 4

Different # of processes, different resolution => M x N /w refinement

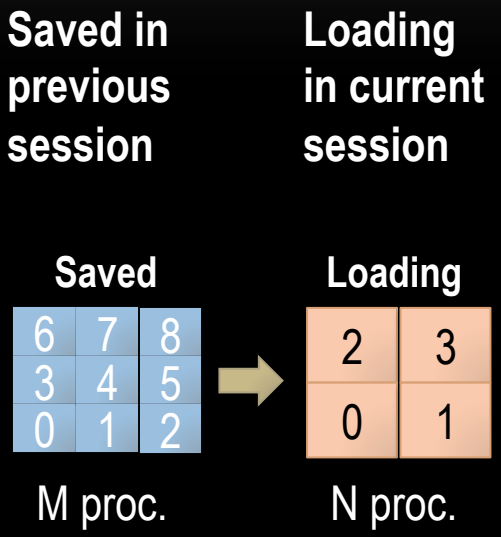


REFINEMENT RESTART

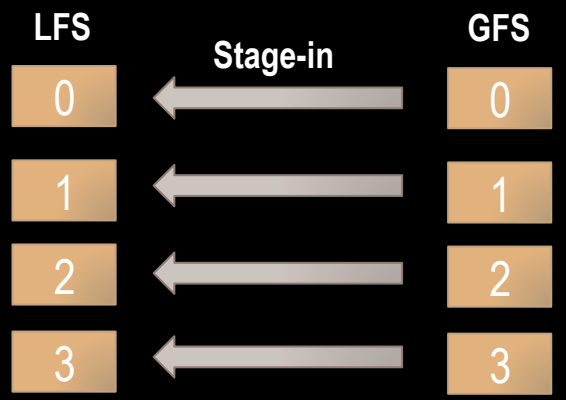
Width (mm)	Model (# Grids)	Nodes (# Process)	Steps	Computed Time (H)	Physical Time (sec.)	Start
16	C2 (0.45G)	9,216	50,000	1.0	2.87	Initial
8	C1 (3.6G)	9,216	20,000	1.0	0.57	Interpolated
4	F (29G)	9,216	10,000	27.4	0.14	Interpolated



M X N /W STAGING




```
N{0} << M{0, 1, 3, 4}
N{1} << M{1, 2, 4, 5}
N{2} << M{3, 4, 6, 7}
N{3} << M{4, 5, 7, 8}
```

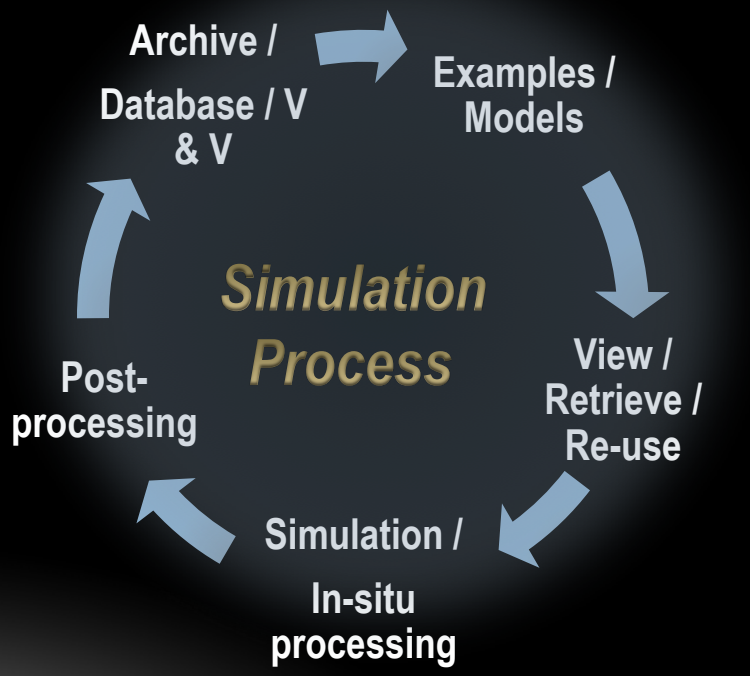


- GFS >> LFSへのリスタートファイルのコピー
- 実行以前に，どのファイルを必要とするかを特定
- ランクディレクトリを用いた転送ができるように準備

PROJECT DATA MANAGEMENT

- Resource management of a project
  - all information; HW info., input files, calculated result files, and derived files
  - Case
    - a unit of execution of a simulation
  - Project
    - a set of cases

- Data management enables us to
  - automatic processing
  - collaboration with database
  - grid search
  - provenance tracking



２．ファイル入出力性能の点から

- 領域分割型の並列シミュレーション

- １プロセスが担当領域の変数について入出力
  - ローカルディスクへの入出力であれば，ベストパフォーマンス
  - GFSの場合には，ファイルシステム依存
  - Lustre >> ブロックサイズ，ストライプ数，．．．多くのつまみがある

- 多数のファイルが生成される問題
  - MPI-IOやHDF5, netCDF, Adiosなどの機能でまとめる
  - File Composition Library (AICS石川T)

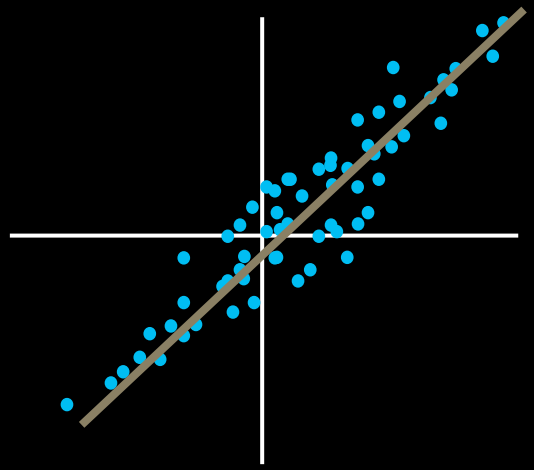
３．データ圧縮

- PROS.
  - ファイルサイズが小さくなる
  - ファイル入出力の時間が少なくなる
- CONS.
  - 万能なアルゴリズムはない
  - 並列化が難しい（エントロピーコーディングなど）
- 方針
  - 様々な状況に利用できる圧縮スキーム群をライブラリとして提供
  - ファイル管理ライブラリと連携して利用

PROPER ORTHOGONAL DECOMPOSITION (POD)

- 行列の固有値計算を利用
- 固有値の大きなものから順に選び，特徴を抽出する
- 高速な固有値計算パッケージが利用可能
- データ復元は，固有ベクトルと係数の線形結合のみ

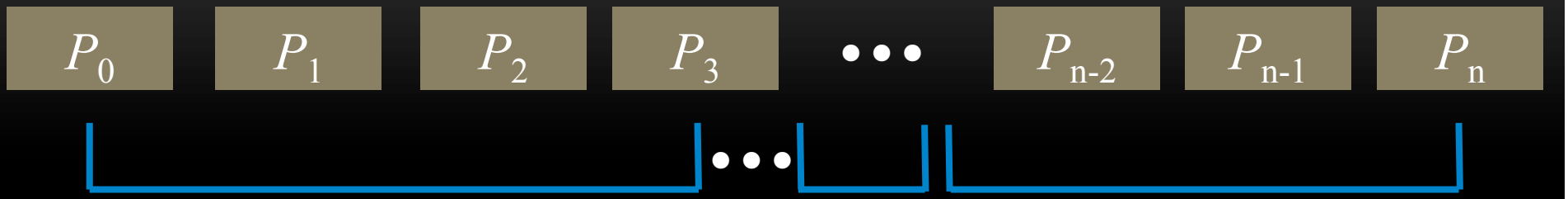
- 逐次計算から並列処理に拡張 [Bi. 2012]
  - 2^n >> non-power of twoへ



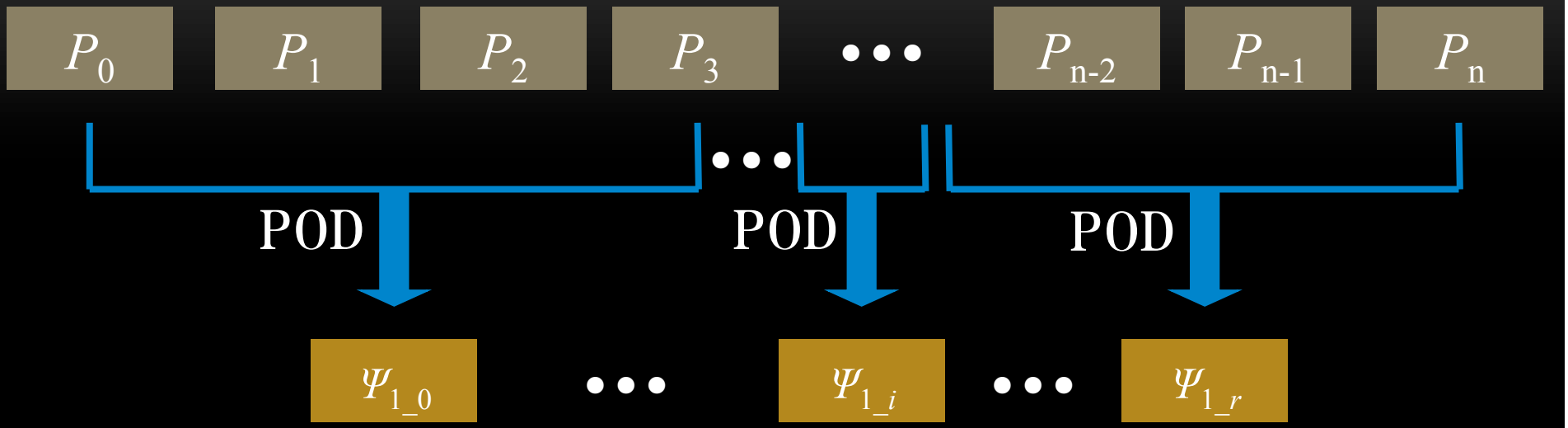
時系列データの圧縮

- 空間規模 ;　　m
- タイムスライス ;　　n
- １タイムスライスの空間データをベクトルとして扱う
- 多数のタイムスライスを行列表現とする $X \in R^{m \times n}$
- 計算量
  - 直交化　$X^T X$　　$O(mn^2)$
  - 固有値と固有ベクトル計算　$O(n^3)$
  - PODの基底計算　　$O(mnr)$
  - 全体のコスト　　$O((mn^2+n^3+mnr) \log_n n)$
- 並列処理では，m, nのパラメータは空間・時間ともに部分を割り当てる

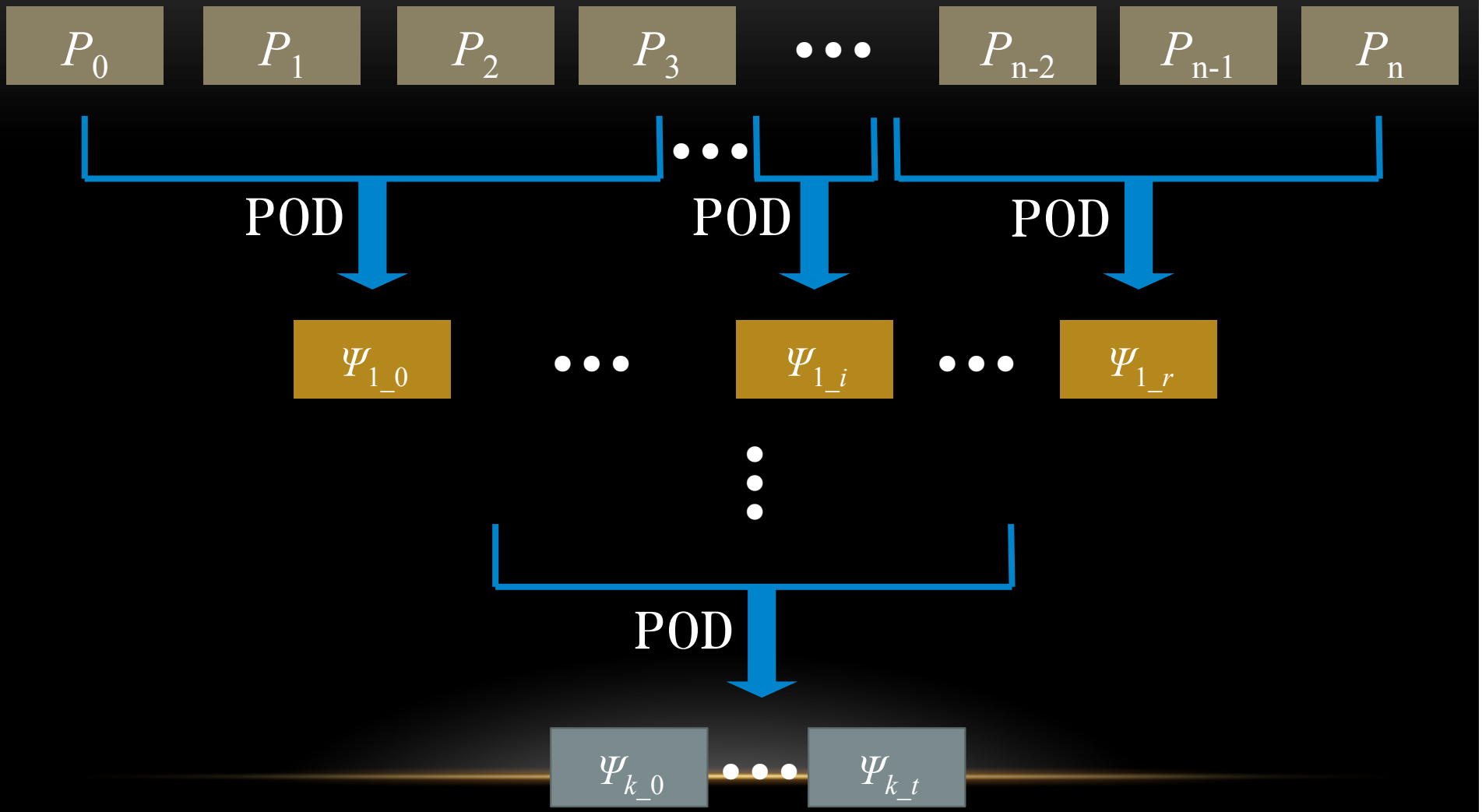
POD IN PARALLEL

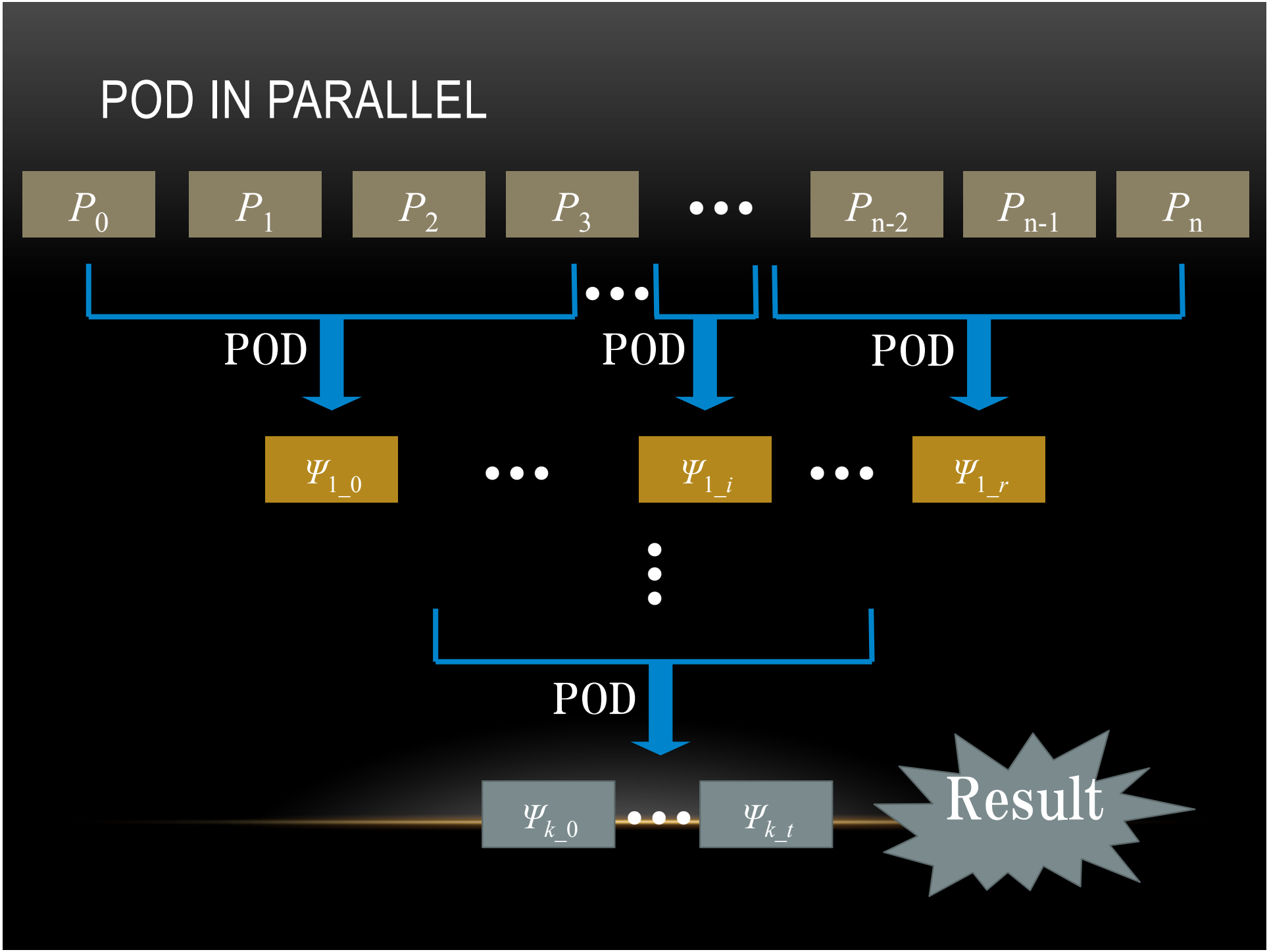


POD IN PARALLEL

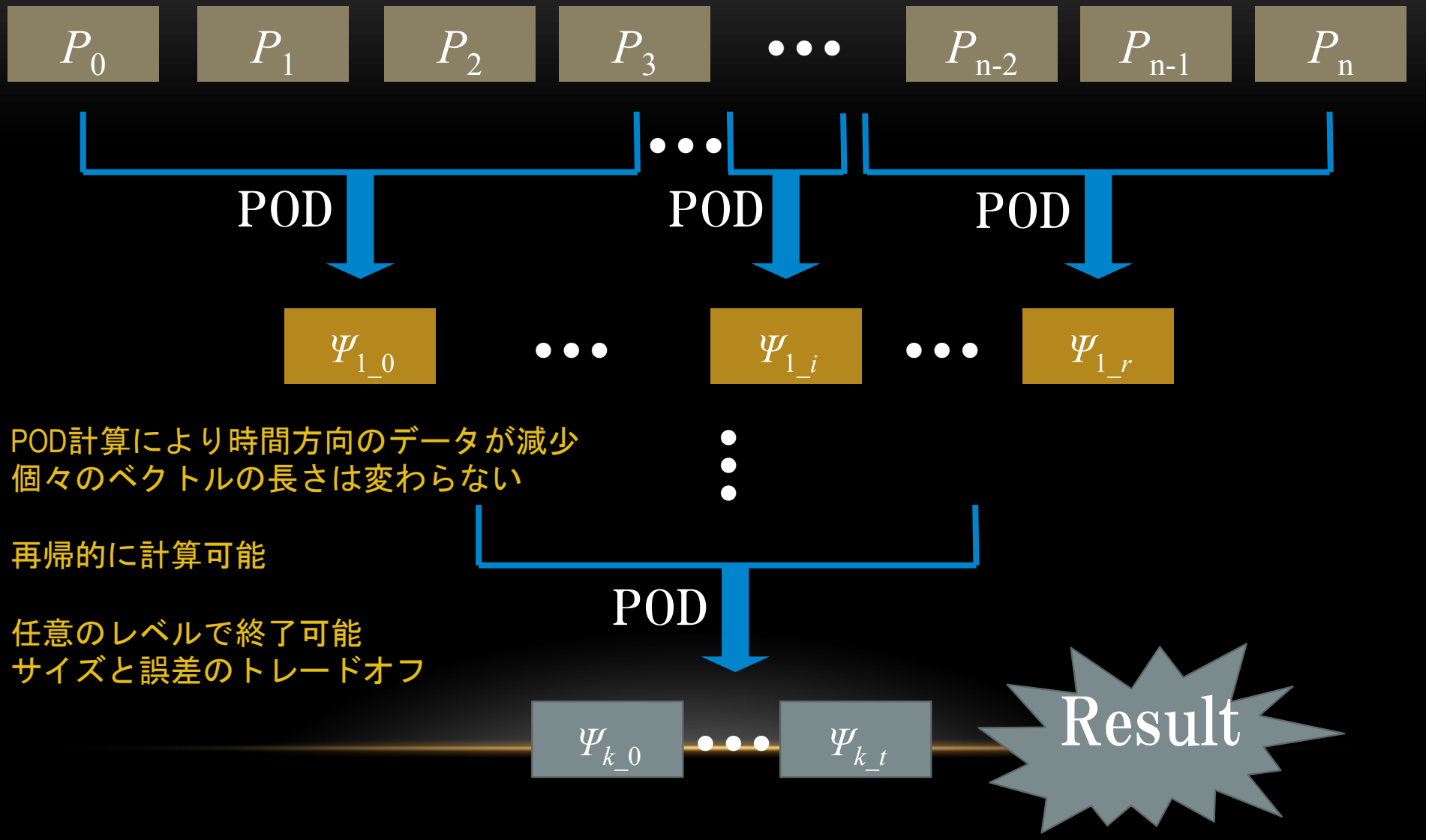


POD IN PARALLEL




POD IN PARALLEL
P_0
P_1
P_2
P_3
P_{n-2}
P_{n-1}
P_n
POD
POD
POD
Ψ_{1_0}
Ψ_{1_i}
Ψ_{1_r}
POD
Ψ_{k_0}
Ψ_{k_t}
Result

POD IN PARALLEL



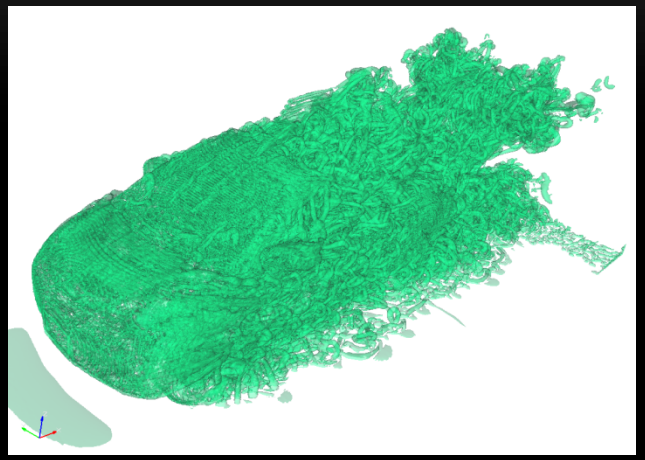
POD計算により時間方向のデータが減少
個々のベクトルの長さは変わらない

再帰的に計算可能

任意のレベルで終了可能
サイズと誤差のトレードオフ

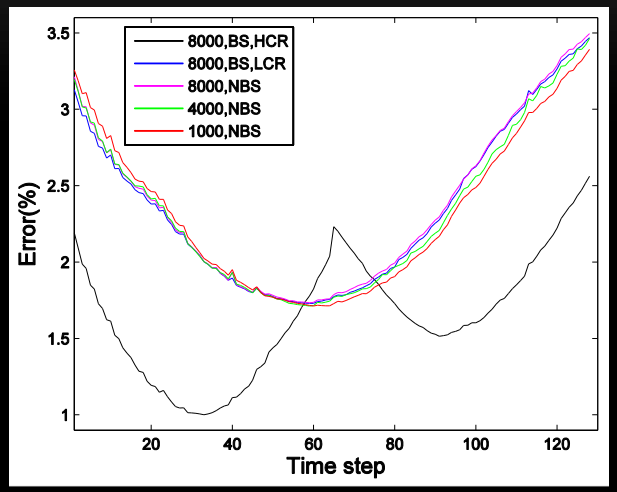
RESULTS – FLOW AROUND A CAR

8000 : 8000 nodes	(N)BS : (No) Binary-Swap	HCR/LCR : High/Low Compression Ratio



Visualizing one timestep

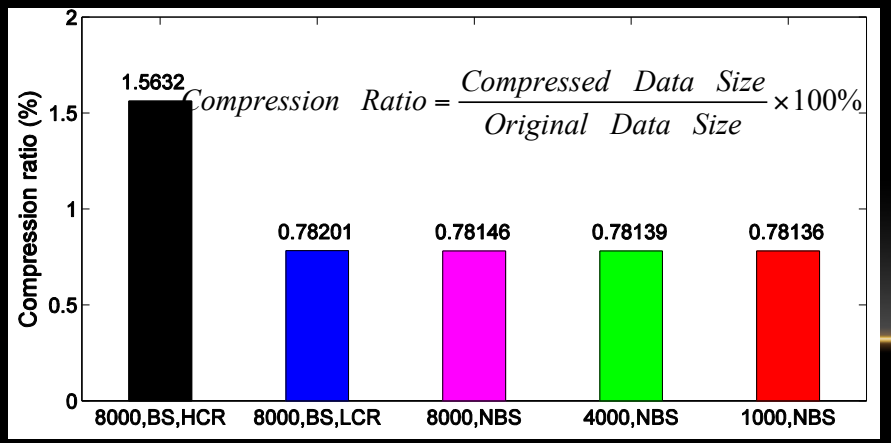
1000 × 400 × 250 × 128 (timesteps)



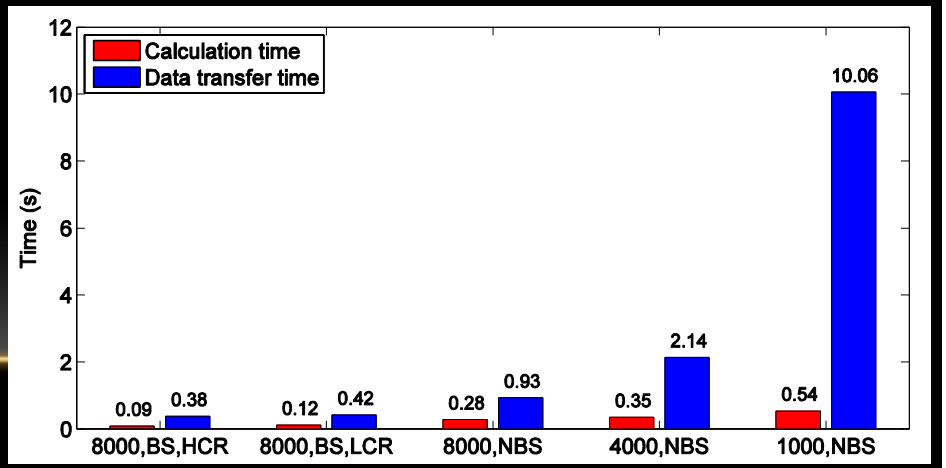
Error

Bad

Good



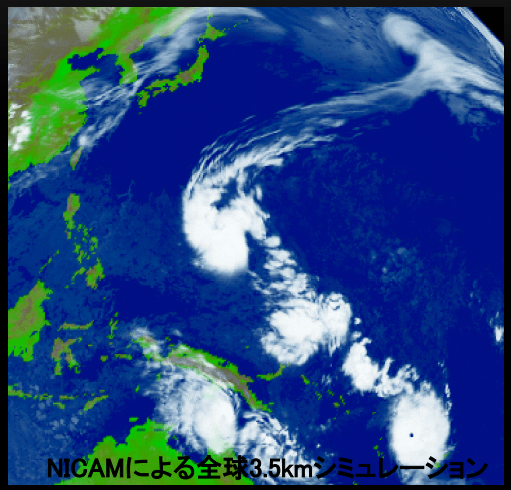
Compression Ratio



Calculation and Transfer Time

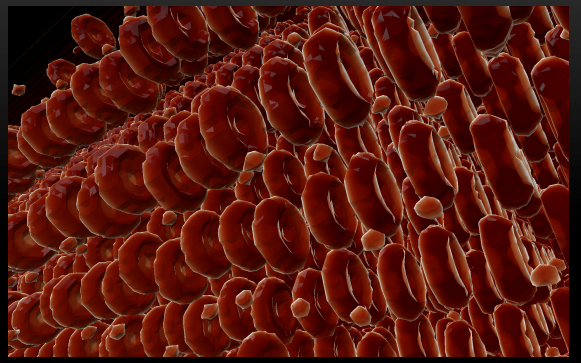
４．大規模データ可視化処理の課題

- データの大規模性
  - 空間規模
  - 非定常データ
  - 多変数

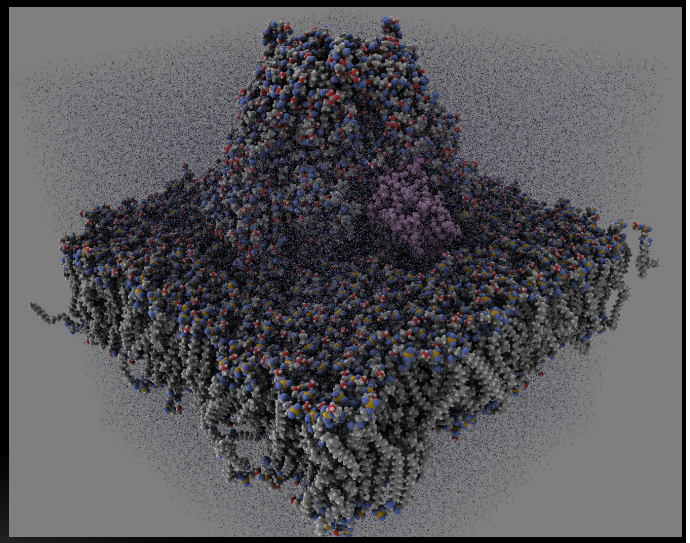


- インタラクティブ性
  - ユーザの操作に対するレスポンス
  - データ量を減らす
    - プリコンピューティング、圧縮、特徴利用、並列処理
  - 効率の良いコード
    - SIMD, アルゴリズム

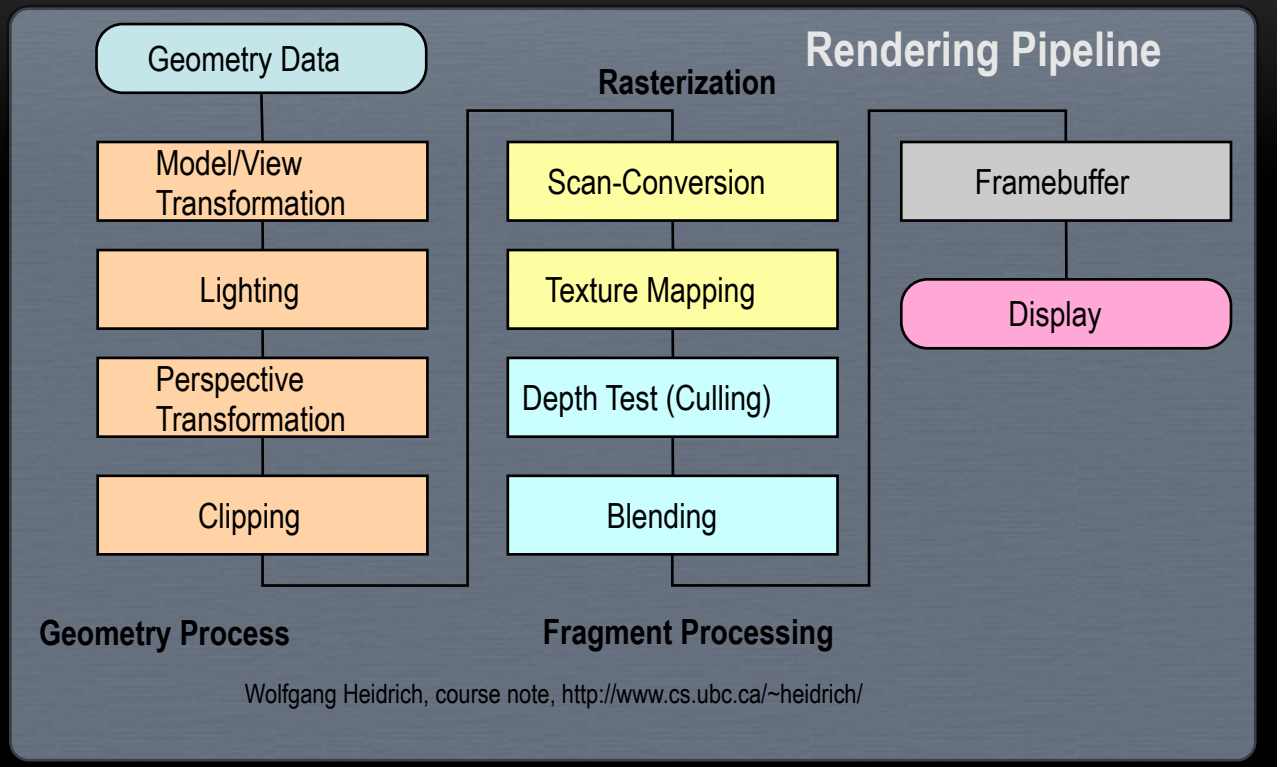
京における可視化



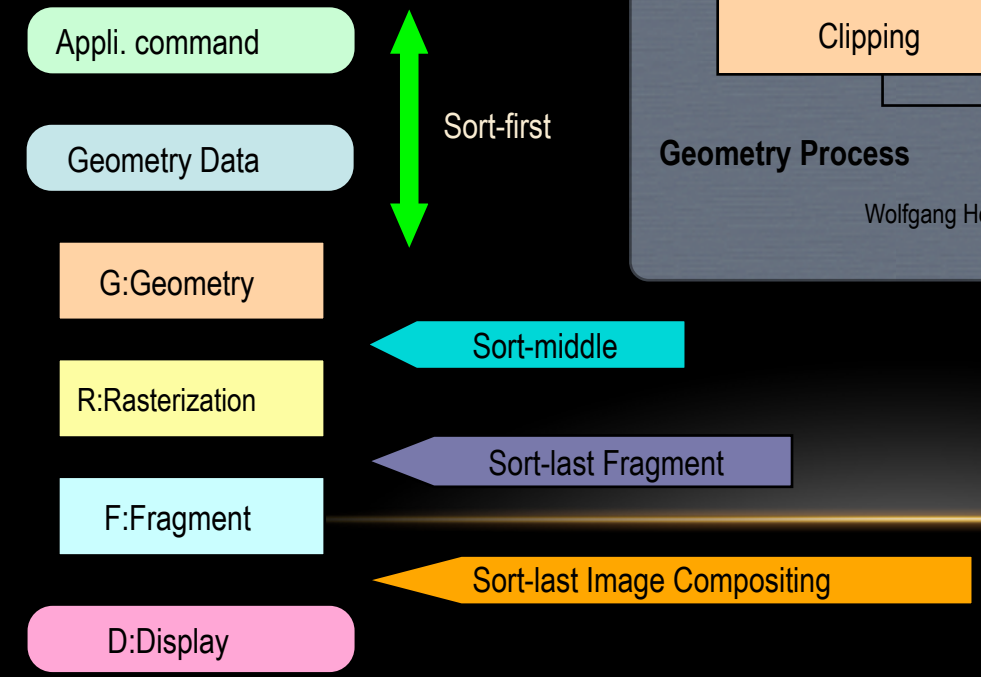
- 大規模な分散並列ファイルを扱う
  - CIO library
- 京の上で直接可視化
  - PC (/w GPU)と動かすコードを共通化 >> GLSL/GLES API, not OpenGL
  - レイトレーサーとボリュームレンダラー
  - ソートラストタイプの並列システム
  - 直交，非構造，点群などのデータ構造
  - とりあえず，バッチ処理
  - インタラクティブ性をどう取り込むかが課題
  - エクサマシンへでも動かすことを考慮



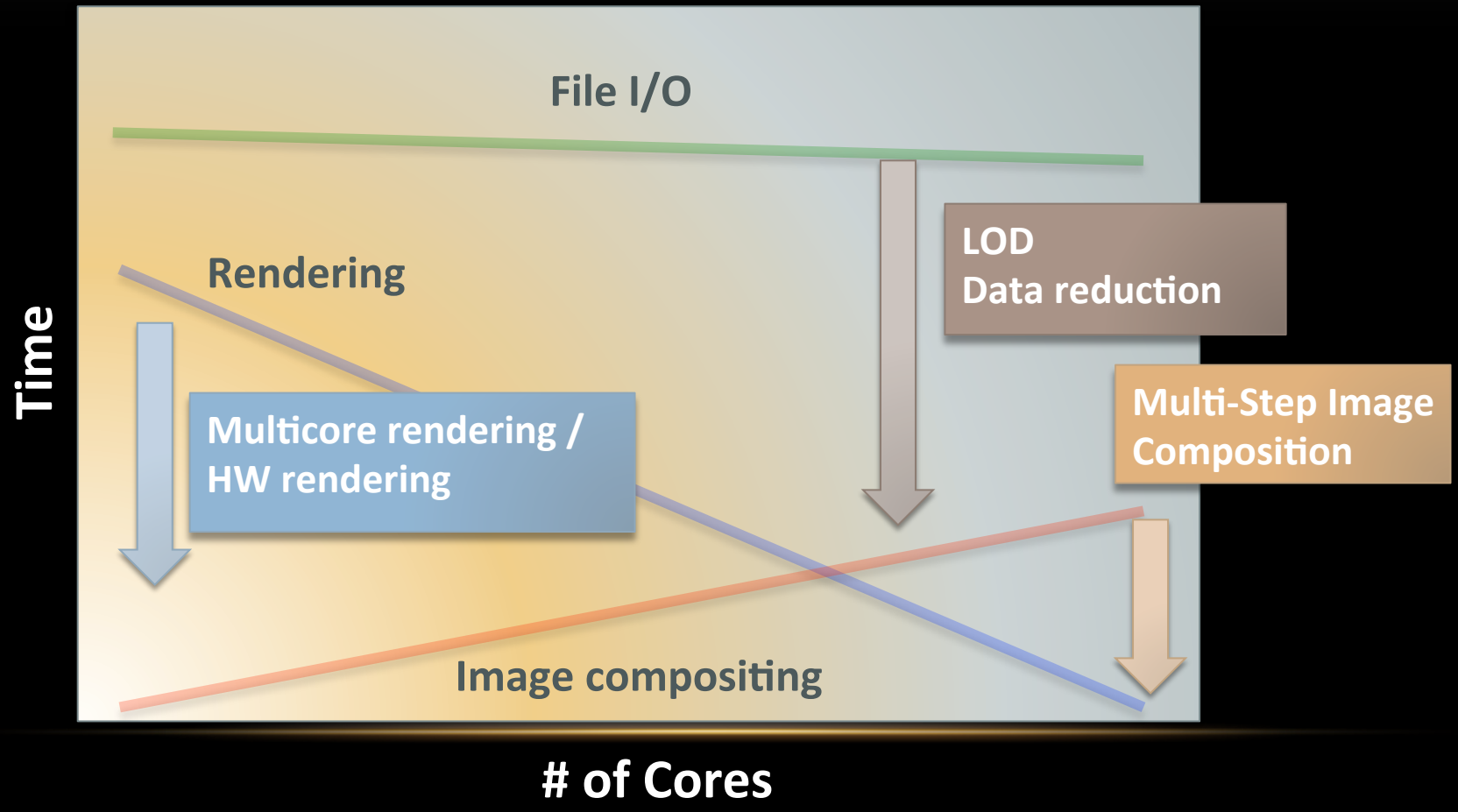
RENDERING PIPELINE



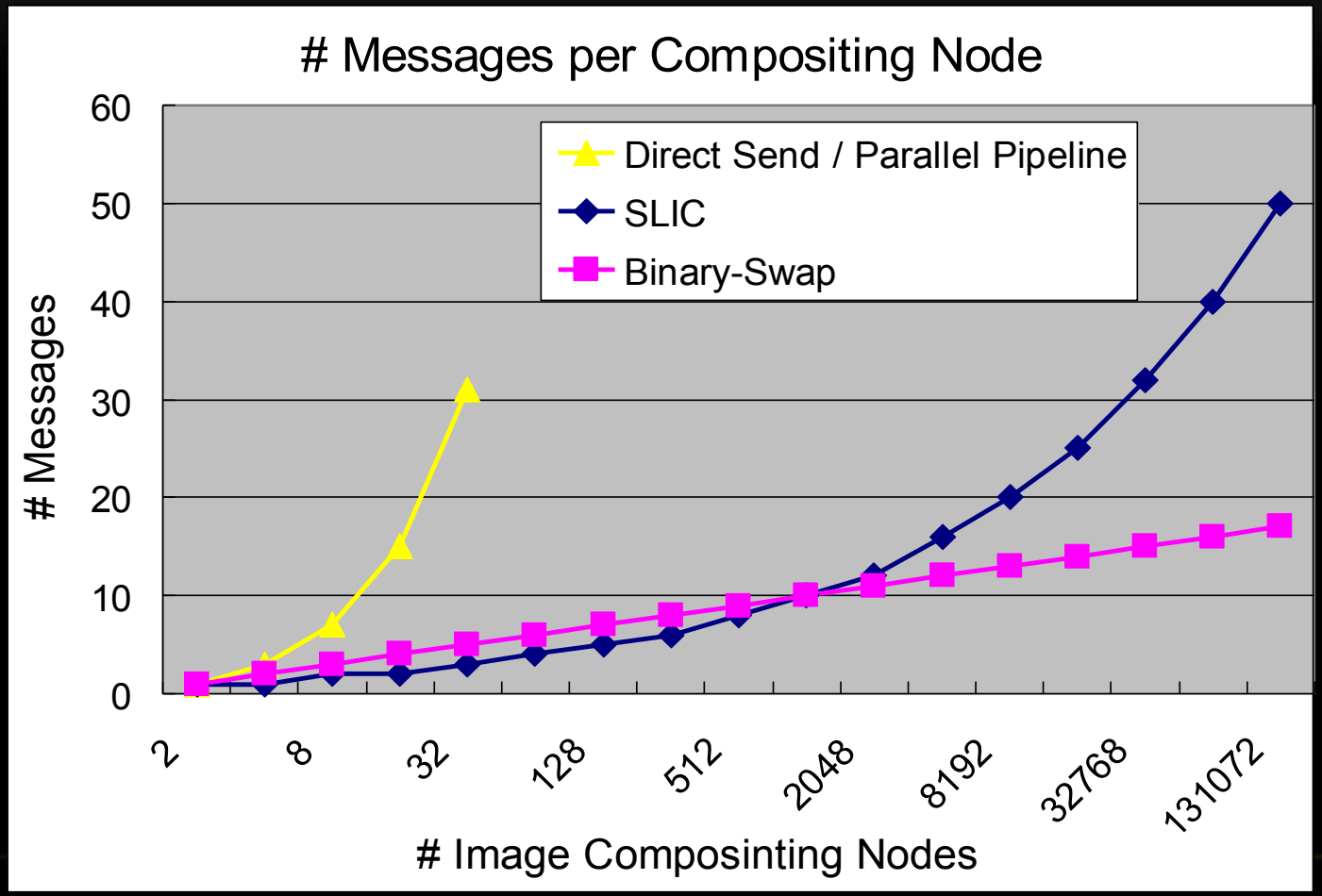
Sorting Taxonomy



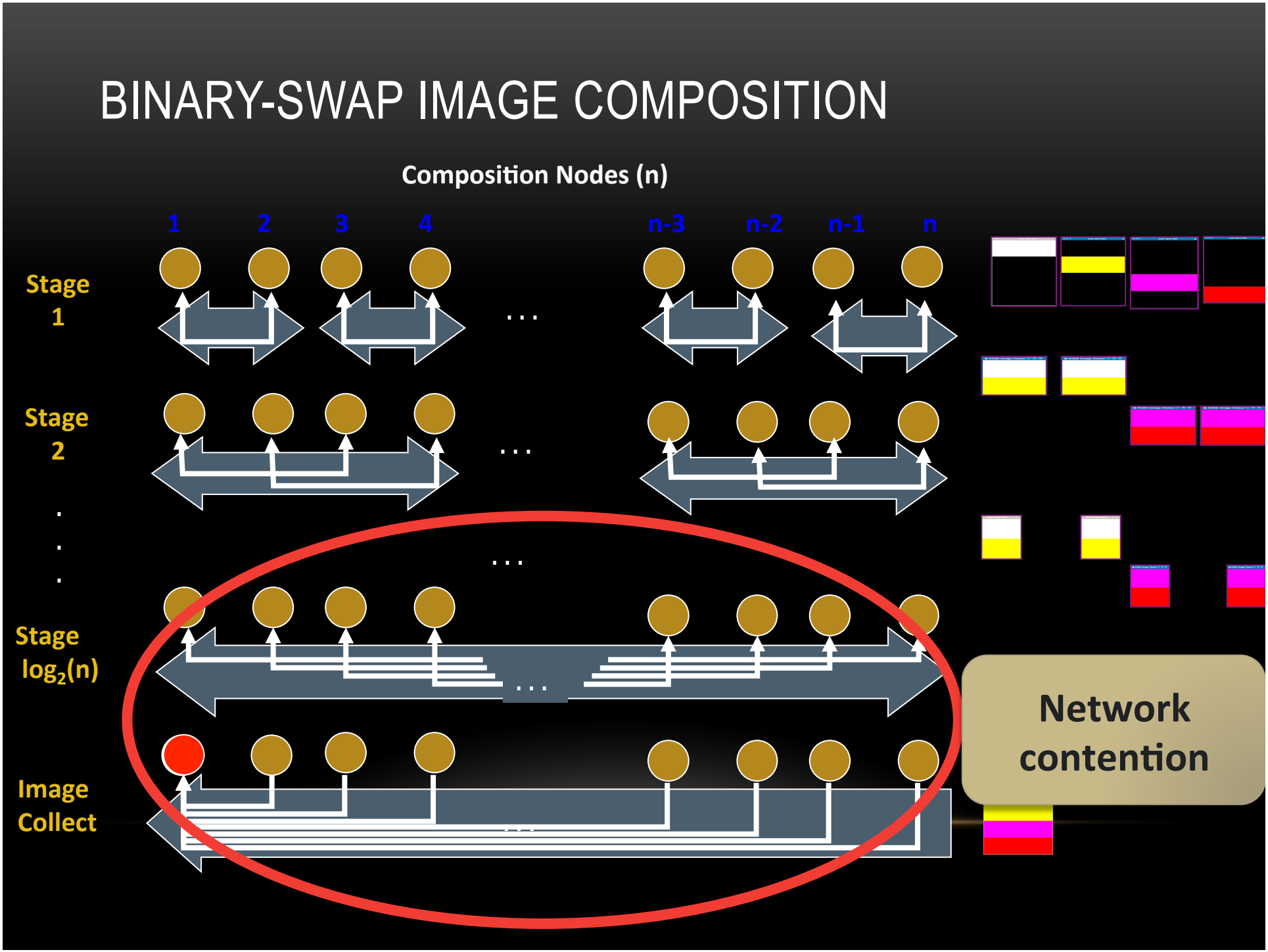
STRATEGY: INTERACTIVITY AND SCALABILITY



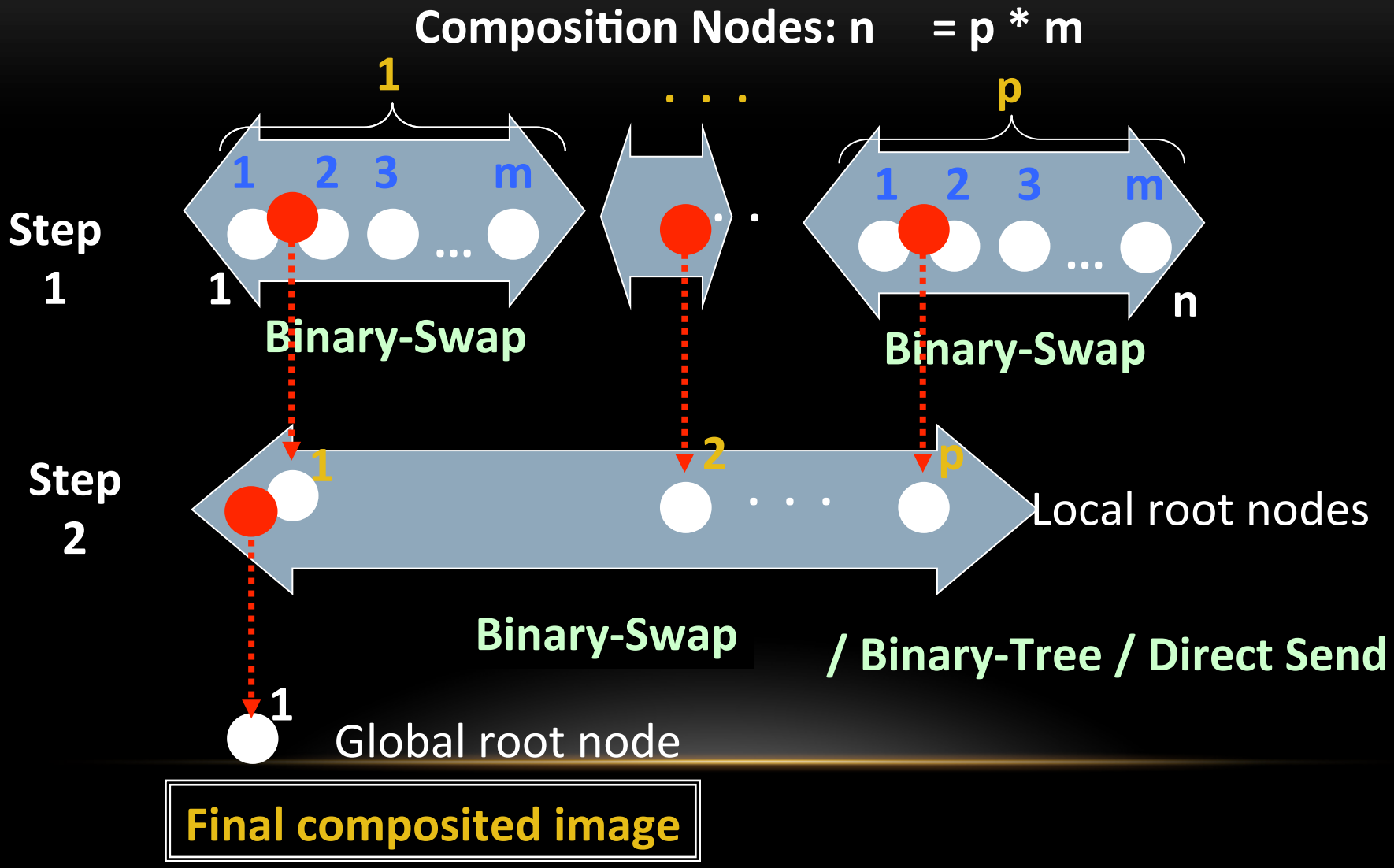
LARGE-SCALE IMAGE COMPOSITING



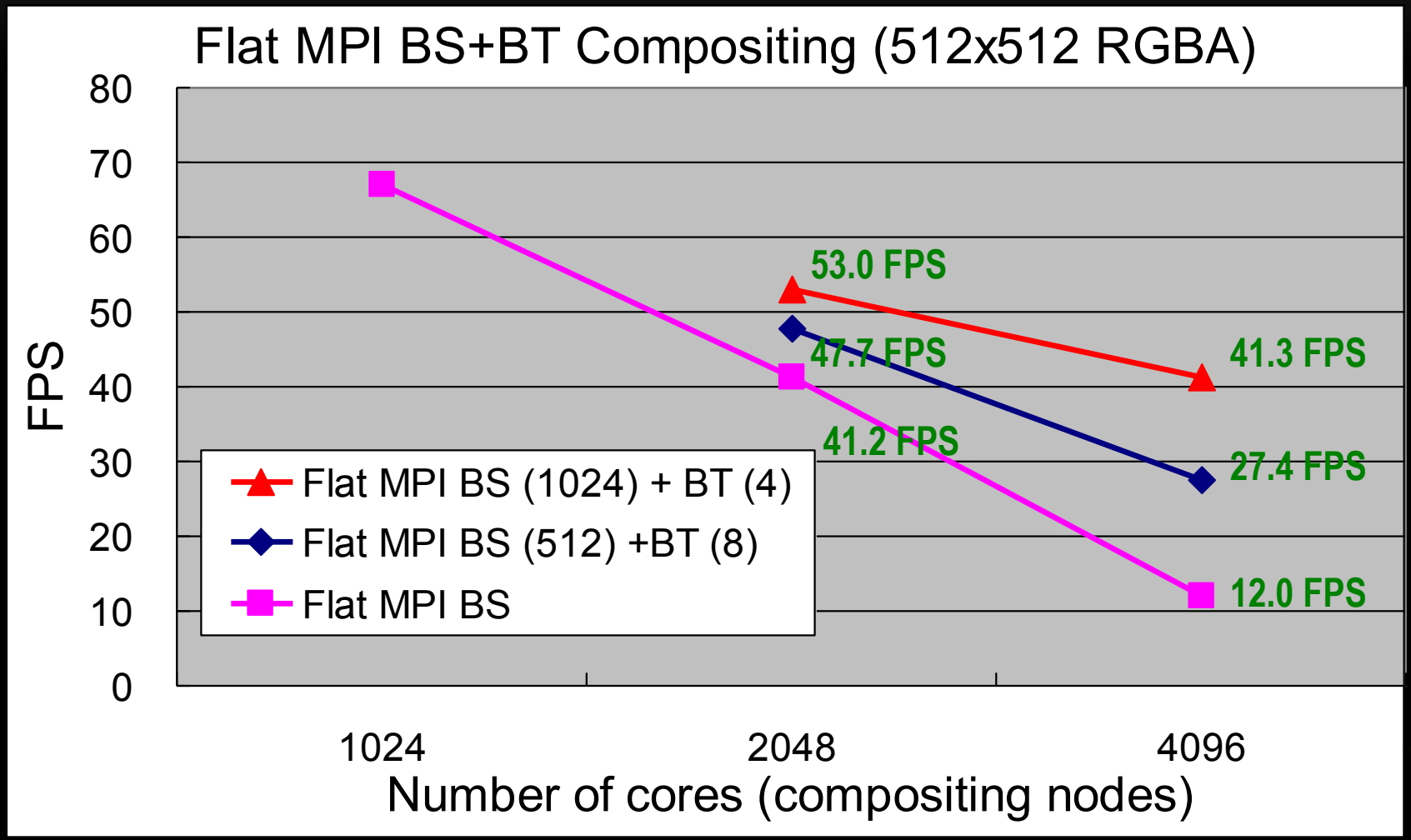
BINARY-SWAP IMAGE COMPOSITION



MULTI-STEP IMAGE COMPOSITION



MULTI-STEP IMAGE COMPOSITING ON T2K



可視化処理の関連技術

- データ処理との融合
  - Visual Analytic
- データベースとの融合
  - クエリー処理
  - ビットマップ処理
- プロセスマイニング
  - 可視化プロセスの効率化
- ワークフロー
  - 定型処理の自動化
  - ポータビリティ
  - 記述性、汎用性

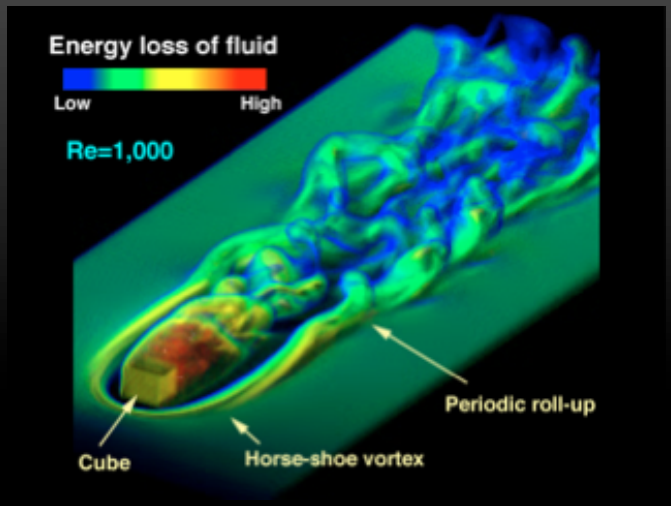
POST PROCESSING

- ファイルベース
  - 多数のファイル管理が必要
  - Lustre >> 並列ファイル入出力
  - ポスト処理の場合データ圧縮は必須
    - losslessとlossyを適材適所で利用
  - データは移動しない．スパコンで計算したら，そこで可視化・分析処理

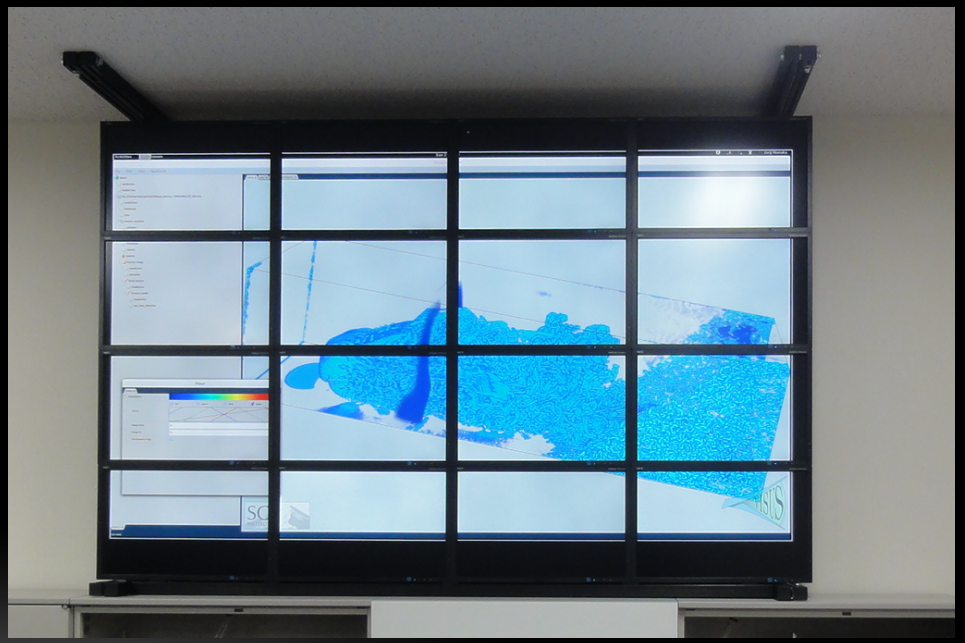
- In-situ へ
  - 計算と同時に分析処理，可視化
  - リアルタイムの統計処理手法
  - シミュレーションと分析処理のロードバランス

どこに向かうか？

- 現象理解，情報提示
- 考える事の支援



- データの中を縦横無尽に探査
- インタラクティブに
- ワークスペース



- イメージと情報の統合
- ディスプレイ技術