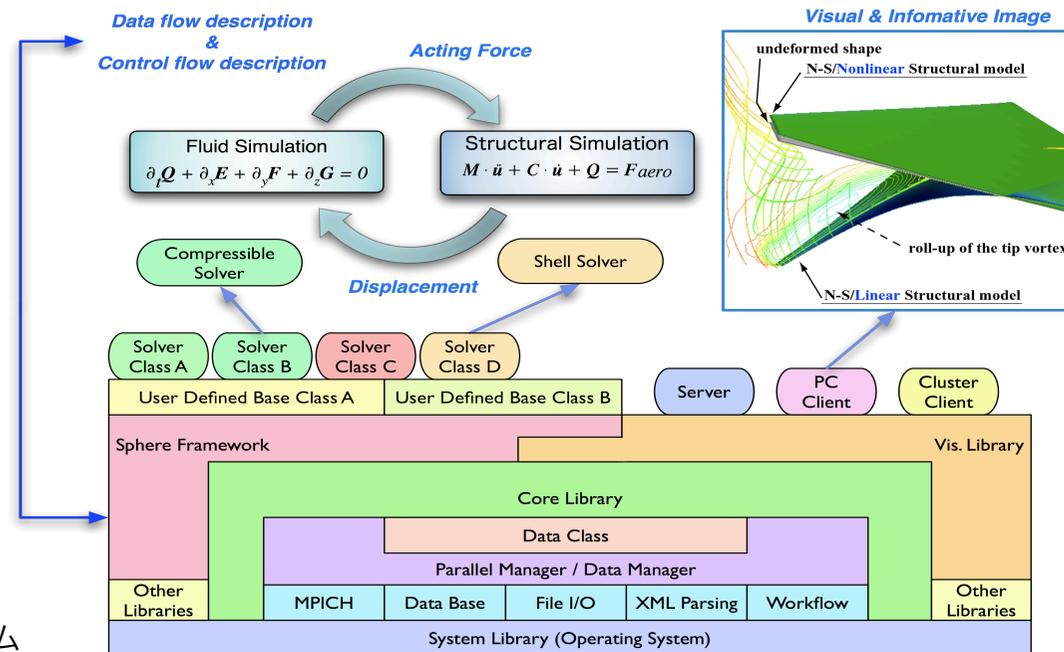


# 次世代スパコンにおける可視化環境

## Visualization Environment on a Next Generation Supercomputer



小野 謙二 keno@riken.jp

独立行政法人 理化学研究所  
VCAD システム研究プログラム  
機能情報シミュレーションチーム



次世代計算科学研究開発プログラム  
生命体基盤ソフトウェア開発・高度化チーム



Riken Symposium, March 13-14th 2008



## Today's Topic

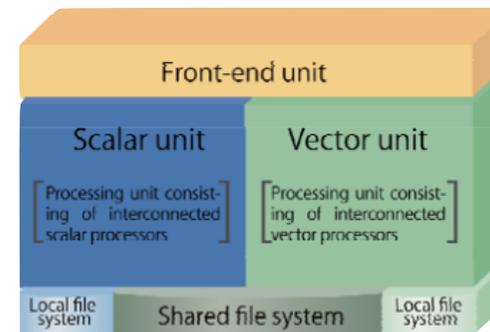


1. 背景と現状の可視化技術
2. 大規模可視化の問題点の定義
3. 目指すシステムのコンセプト
4. 開発の経緯
5. 開発スケジュール
6. 開発システムの詳細

# 可視化システム開発の背景

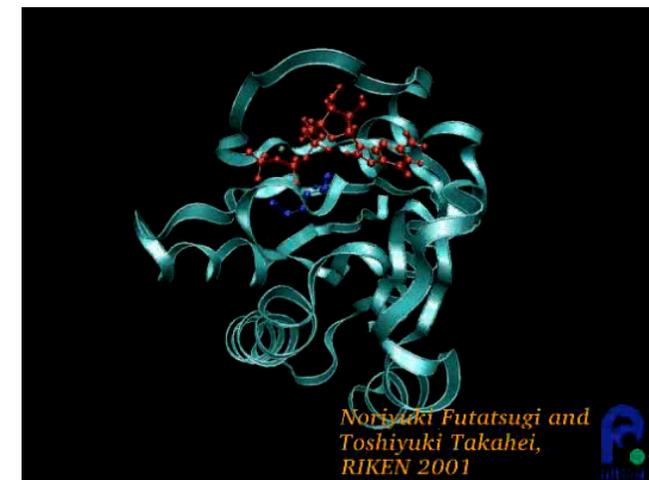
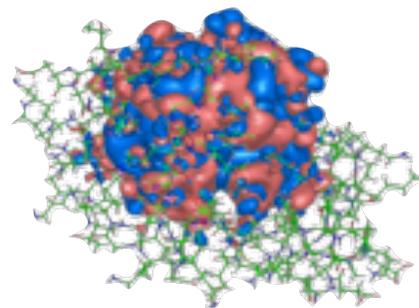
## ■ 次世代計算機開発プロジェクト

- ハード・ソフトの新規開発
  - ✓ ベクトル・スカラ融合の超並列計算機
- 開発課題
  - ✓ CPU, ネットワーク, ファイルシステム, OS, コンパイラ, . . .
  - ✓ 性能, スケーラビリティ, 運用, ユーザビリティ



## ■ 適用分野

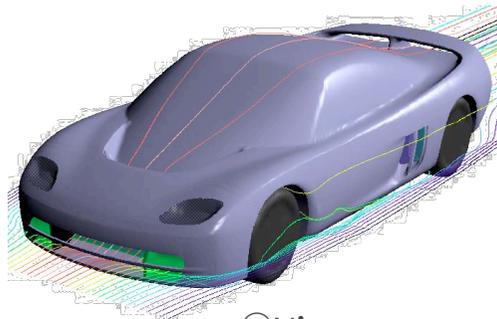
- 主要8分野 ライフサイエンス分野に注力
- Science, Engineering



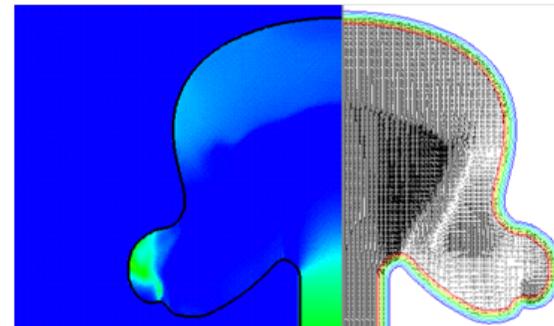
# 対象とする多様なシミュレータ

## 狙い

- 開発期間短縮, コスト低減
- 品質向上
- 現象解析



©Nissan



## シミュレータ

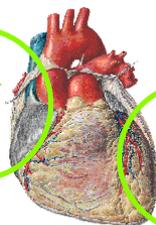
- 熱, 構造, 流体, 電磁波, 振動, . . . .
- マルチスケール, マルチフィジックスヘシフト

m-scale



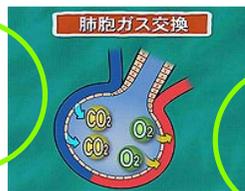
Cardiovascular

mm-scale



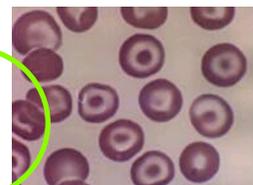
Organ

$\mu$ m-scale

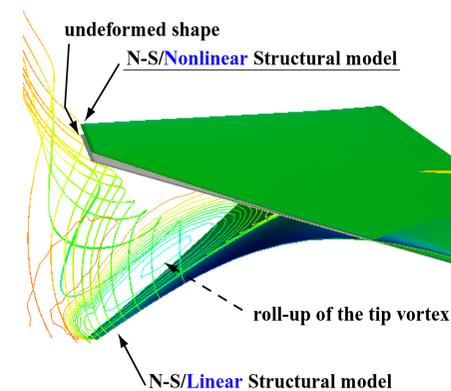


Material transport

nm-scale



BRC, PLt



## 可視化に求められること

### ■ マルチフィジックス解析

- 複合的な物理現象のインタラクションの理解

### ■ マルチスケール解析

- 異なる時空間的スケール現象の描像

### ■ 並列アプリケーション

- 多数のファイル群のハンドリング

### ■ 生産性の向上

- リモート時の高レスポンス
- データマネージメント機能
- 様々な可視化・分析機能

### ■ 使いやすいソフトウェア

- ユーザーインターフェイスの統一
- カスタマイズ性

# 可視化システムの現状

## ■ 商用

- AVS, Field View, Ensight, ...

特徴が異なるため、棲み分け可能

## ■ 非商用

- VisIt LLNL, open source, 並列可視化アプリ
- ParaView LANL, open source, 並列可視化アプリ
- VTK 汎用可視化ツールキット (開発環境)
- RVSLIB NEC 可視化ライブラリ
- VisLink 富士通 リモート可視化
- VisTrace 富士通-JAXA
- Naregi 分子シミュレーションのリモート可視化
- Postkunj Jaxa, 東洋大
- ...

積極的な  
成果の利用

# 理研：次世代計算科学プロジェクト

## ■ 目標

- 次世代スパコンで稼働するアプリの開発・高度化支援および実行支援

## ■ 生命体基盤ソフトウェア開発・高度化チーム

- アプリ・ミドルウェア
  - ▶ ライブラリ，フレームワーク利用による高性能コード開発の効率化
  - ▶ ターゲットは，高並列，連成ソルバ
- 大規模データ可視化システム
  - ▶ リモート環境での可視化機能
  - ▶ イメージだけでなく，統計処理や知的データマイニング機能も含む

# 大規模可視化システム開発の問題点

## ■ 大規模データと並列環境

- MMU, Network, HDDの容量と速度性能の階層性を考慮
- 並列処理時の性能

## ■ ユーザの多様な可視化シナリオに対応するサービスの提供

- 扱うデータ規模 (PB, TB, GB) に応じたデータハンドリング
- リアルタイム/ポスト, バッチ/インタラクティブ, リモート/ローカル

## ■ 運用・システム展開

- システムの運用形態依存の問題
- NISや他の並列システムへの適用

## ■ スケジュール

- アプリ開発段階での利用 > 早期からのリリース

# 目指すシステムのコンセプト

## ■ シミュレーション結果から、有用な知識・情報を効率的に引き出す

- 様々な可視化手法による場の把握
- 生成された2次データの利用
  - ✓ イメージ, 分析結果
- 多くのデータの管理のしやすさ

## ■ データ管理

- シミュレーション開始から分析後までの全てのデータ
- 個々のデータの関連づけ

## ■ 処理の効率化

- 定型作業の自動処理

大規模可視化システム

考える過程を支援  
Scientific Data  
Management

データベース

ワークフロー

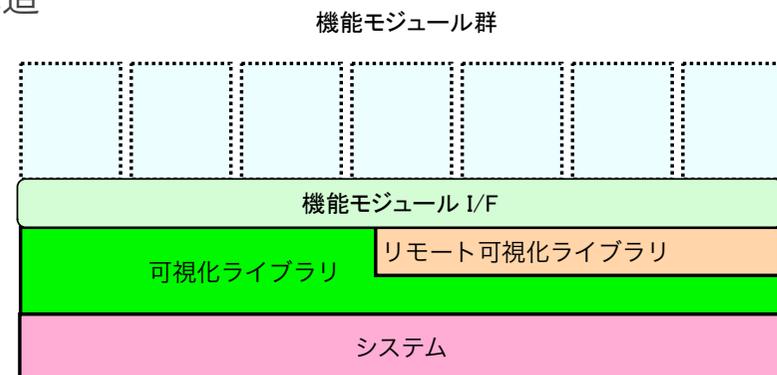
シミュレーションの価値を増幅させる, 知識を得るためのツール群

# 可視化システムの特徴/要件

- リモート可視化とローカル可視化  
共通クライアントによる統一環境の提供
- リアルタイム可視化とポスト可視化  
ファイル経由の可視化を原則
- インタラクティブ可視化とバッチ可視化
- SWレンダリングとHWレンダリング
- 大規模データハンドリング
- 並列可視化
- 可視化基本ライブラリと機能モジュール群の構造  
将来の機能拡張と実装の容易性を担保
- 移植性  
Linux, Windows, Max OSX, PC cluster, Supercomputer
- 市販可視化ソフトとの連携  
リモート可視化プロトコルの共通化

ユーザの視点に立脚した利便性の追求

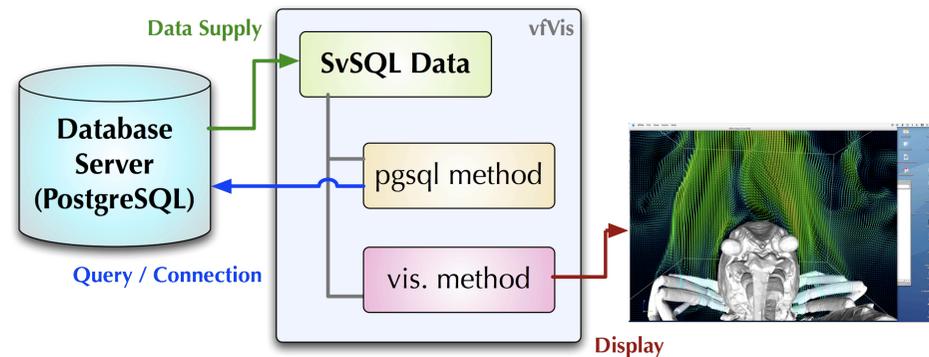
多様な要求に応える基本設計



# データマネジメント機能

## ■ Scientific Data management

- ファイル管理とアプリ（ライブラリ）内部でのオブジェクト管理に利用. あるいは, 外部からのクエリに対するサービス
  - ✓ 計算結果
  - ✓ パラメータ
  - ✓ ジオメトリデータ
  - ✓ メタデータ
  - ✓ イメージ
  - ✓ ファイル
  - ✓ スクリプト
  - ✓ 生データと加工データの関連付け
- Database
- Workflow



## 開発方針と経緯

- エンドユーザに対するヒアリング
- ユーザシナリオの抽出
- ユーザシナリオを満たすシステム構成の検討
- システム構成案
  - 基本システム + 拡張可能な機能モジュール群
  - 基本システム
    - ✓ 大規模データのハンドリング機能
    - ✓ 並列描画, 画像重畳機能
    - ✓ モジュール群へのインターフェイス
  - 機能モジュール群
    - ✓ 個別の要素技術をモジュール化, 基本システムにプラグイン

## ユーザ要求の一例（調査結果から）

### ■ ユーザアプリケーションへの対応

- 多様なデータ構造
- 利用方法も様々

### ■ 解析の状況把握とパラメータ変更

- モニタとステアリング
- 計算と同時に結果を観察（リアルタイム）

### ■ インタラクティブ性

### ■ 後処理で分析

- バッチ処理
- 比較可視化, 定型可視化, データ分析

### ■ 手元での可視化処理が圧倒的多数

- 使い慣れている可視化ソフトの利用

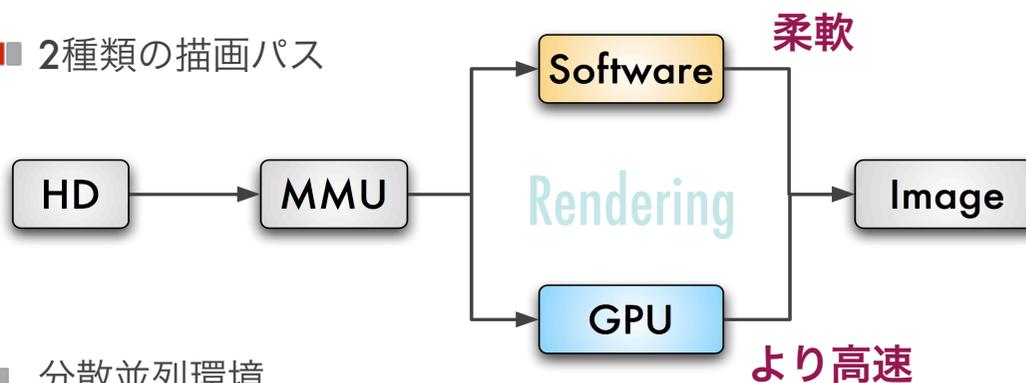
# 可視化

- データを人の目により認識し，理解力を促進する
- 多角度・側面からの検討により，本質の理解を助ける



- インタラクティブ性（スピード）とUI（作業効率）が重要な鍵
- リモート環境でも，5fps程度の性能が必要

- 2種類の描画パス

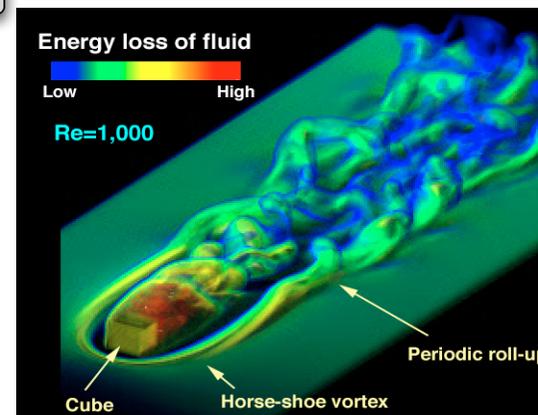


- 分散並列環境

- レンダリングパイプラインの効率的な並列処理
- イメージ/データ通信のための高速ネットワーク

高品質イメージは認識能力を補助

インタラクティブ性は空間認識力を高める



## ソフトウェア (CPU) 描画

- 一般解

- 柔軟なCPU処理による高品質イメージ生成
- 描画性能が低い
- 他のシステムへの水平展開が容易

- インタラクティブ性の確保のためには

- 描画処理 (パイプライン) の並列化
- 並列CPU (マルチコア) 活用技術
- 高速なレンダリングアルゴリズムの採用
- 十分な帯域のノード間通信能力とイメージ重畳アルゴリズム
- 扱うデータ量の削減

- 検討項目

- 既存の可視化システムとの違い > 新規性 (知的可視化技術)

## ハードウェア (GPU) 描画

- 特殊解

- 高速で高品位な表示が可能 > ソフトの数十倍以上の性能
- GPGPU (nVIDIA CUDA, AMD FUSION)技術の恩恵 > コプロセッサ能力

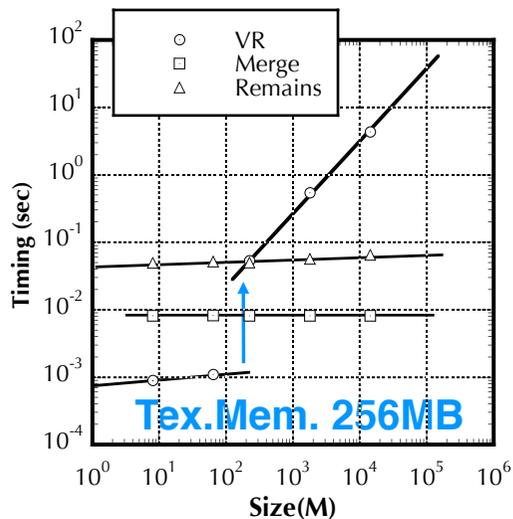
- 検討項目

- システムと適合するドライバが必要, ドライバ開発の場合, ハードウェア内部情報が必要なため開発元との協力が必須
- GPUの有効利用のため, 主記憶からの十分なデータ転送能力が必要
- プログラミングモデルがCPUとは異なる
  - ✓ 最新技術のキャッチアップ
    - ⇒ インコア処理が高性能の前提条件. 前処理・データ削減は必要
- ハード動向ウォッチ
  - ✓ GPUリソースのTSS的利用の可能性 > GPU context switch
  - ✓ バスはPCI-Express 16xで十分か? HTの可能性もあり
- コストと消費電力

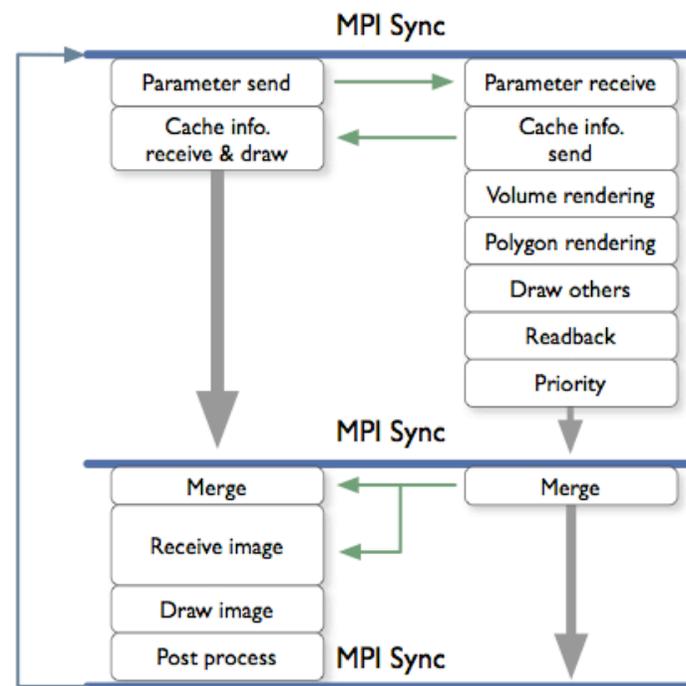
# 並列GPU可視化のポイント

## GPUレンダリング

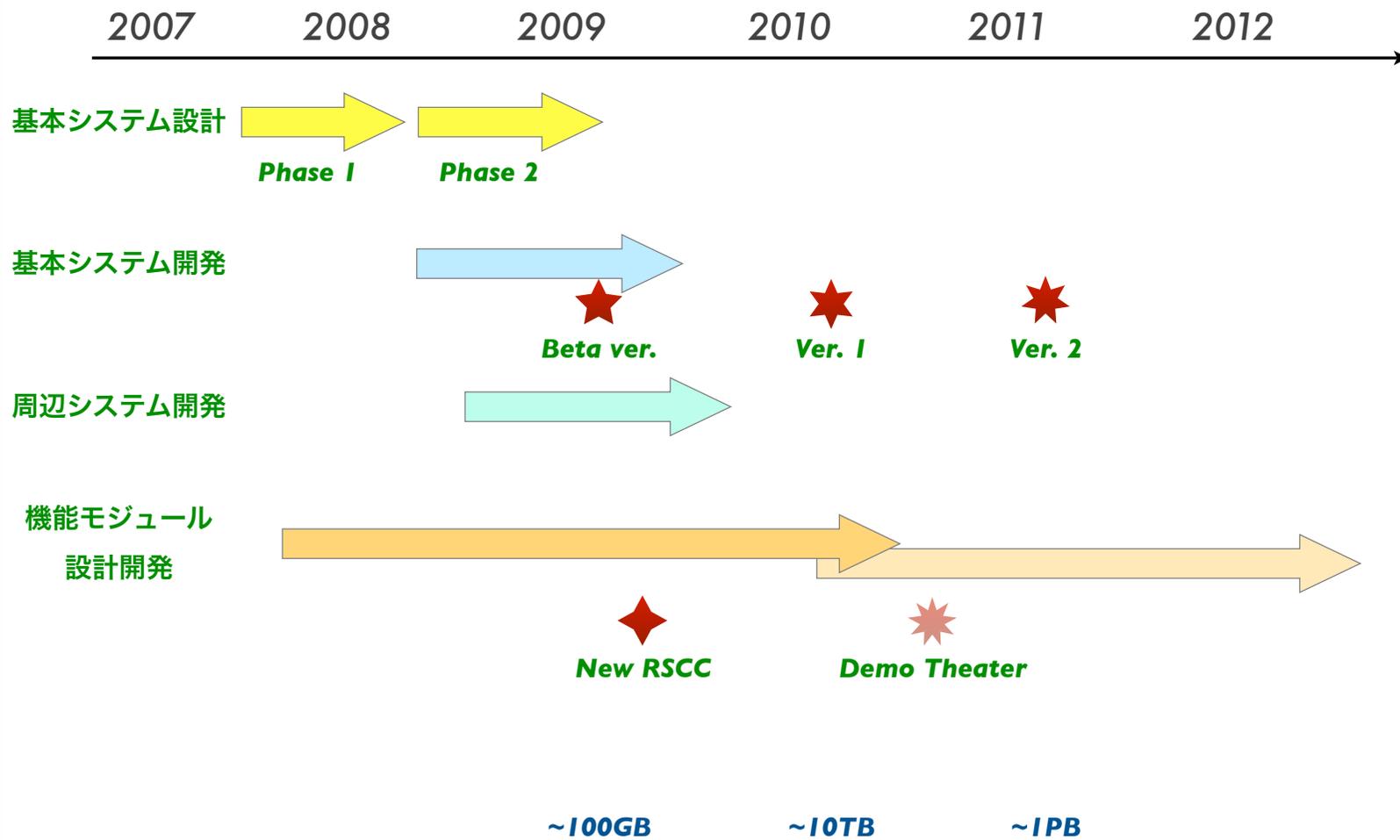
- Tex.Mem. ~数GB << MMU
- インコアの時のみ高性能
- 全データに対する描画は、ダウンロードの繰り返し
- 高い描画性能のためには、データ圧縮（前処理）の必要
- 描画性能はレンダリングコードの実装依存



- 領域分割された各領域のイメージを重畳する効率的なスケジューリング
  - BSP (Binary Swap Partitioning) Tree
  - SLIC (Scheduled Linear Image Compositing)
  - FPS = 1 / (Rendering + Reedback + Merge + Utility + Barrier)



# スケジュール



## 可視化システムの詳細 - keyword

- 可視化基本機能
- インテリジェントデータリダクション機能
- データマネジメント機能
- ファイルフォーマット設計
- マルチプラットフォーム対応クライアント
- リモート機能, バッチ機能
- 並列レンダリング
- 知的可視化機能
- データ分析, マイニング機能
- インタラクティブ機能
- ライブラリ機能
- パラメータスイープ
- 先進レンダリング機能
- **Time-Varying**データハンドリング機能
- デバッグ機能
- デモ・シアター

必須機能:  
基本システム機能  
+  
ベーシック機能モジュール

大規模データ可視化の特徴的機能

先進的・魅力的な機能

オプション (RSCC, 神戸)

## 機能モジュール

- 基本可視化メソッド
- インテリジェント・データリダクション
- データマネジメント
- パラメータスイープ
- 知的可視化機能
- 統計処理
- デバッグ
- バッチ
- アノテーション
- シナリオオーサリング
- 立体視



## 基本可視化機能

### ■ スカラ

- コンタ, ペイント, VR, 等値面

### ■ ベクトル

- ベクトル図, 流線, タイムライン, 流跡線, LIC

### ■ データプローブ, グラフ

- 値の表示

### ■ オブジェクト生成

- 等値面ポリゴン, グリフ

### ■ ビュー・ウィンドウ操作系

- UI, ウィザード

# インテリジェントデータリダクション機能

## ■ データ圧縮技術

- 符号処理による圧縮・伸長
- 量子化
- 間引き
- LOD技術

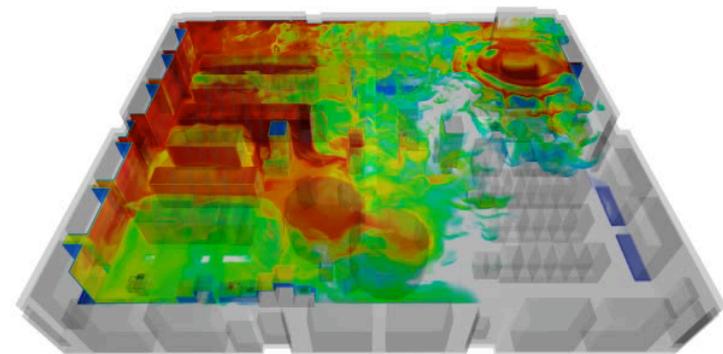
## ■ 観察対象となるデータの抽出

- 領域選択 (ROI, Region Of Interest) , ROIの移動 (Roaming)
- 特徴抽出 (Feature Extraction) により, 小規模のメタデータに変換

## ■ データベースの利用

- パラメータ並列の分析作業時のデータ管理
- 可視化結果の管理
- 生データとメタデータとの関連づけ

## ■ インタラクティブ可視化の前処理の機能も提供



# 知的可視化機能

## ■ 特徴抽出

- 渦管, 特異点

## ■ アノテーション

## ■ 比較

## ■ Illustrative rendering

## ■ グリフ

## ■ メタデータ化

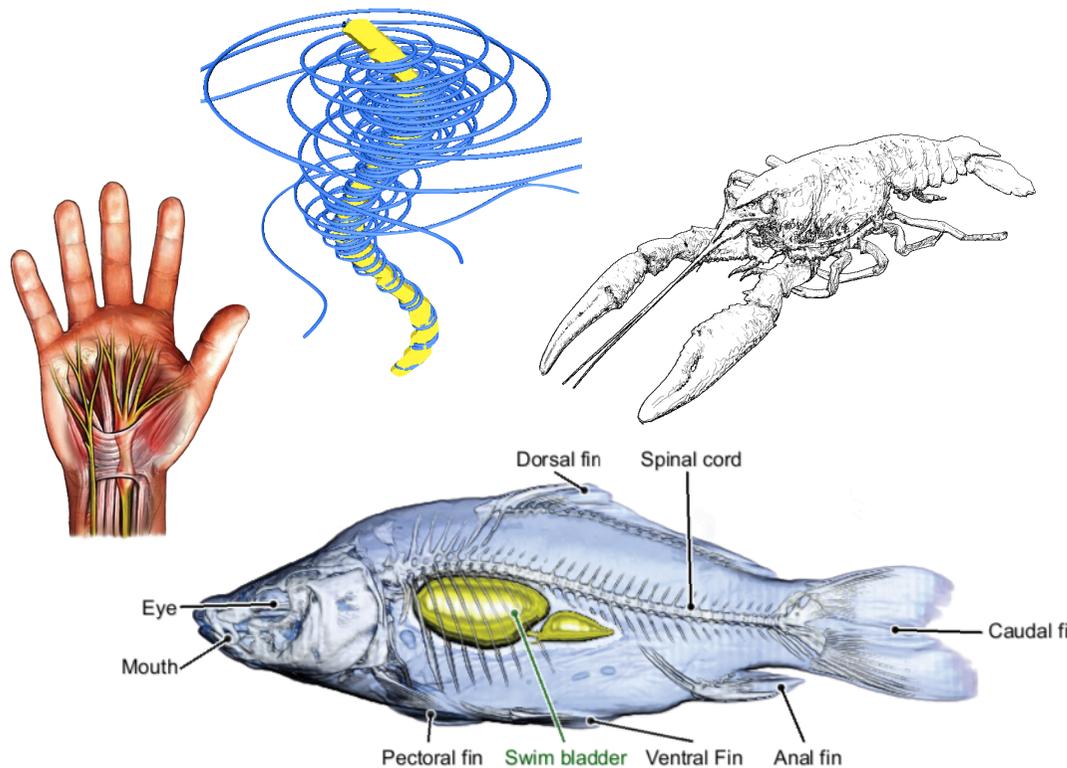
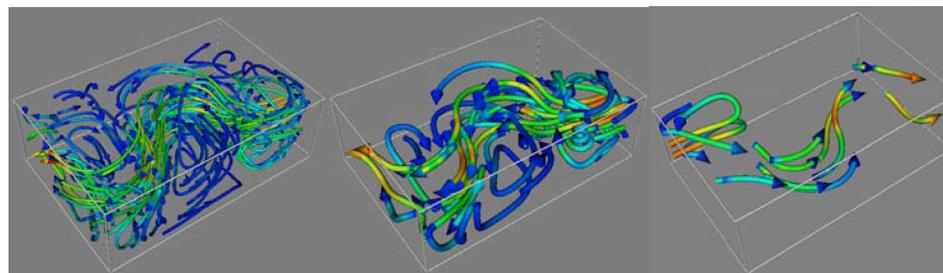
## ■ 手順の自動化

## ■ データに対する演算

## ■ 推奨パラメータ抽出

## ■ より具体的な手法は問題依存

- 研究者と協同で



# 先進レンダリング機能

## ■ 高品位レンダリング

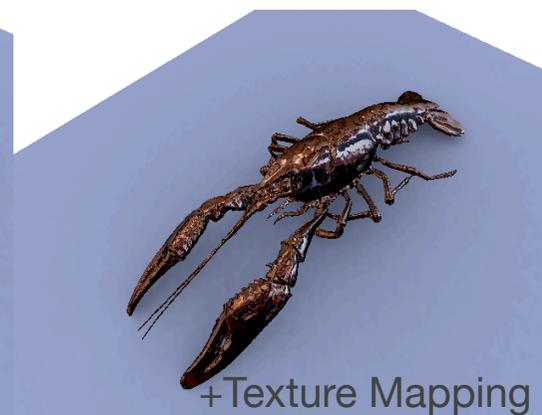
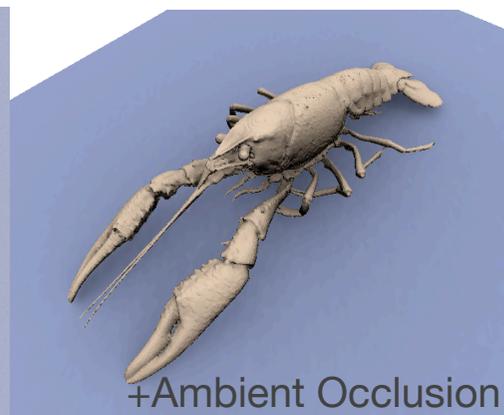
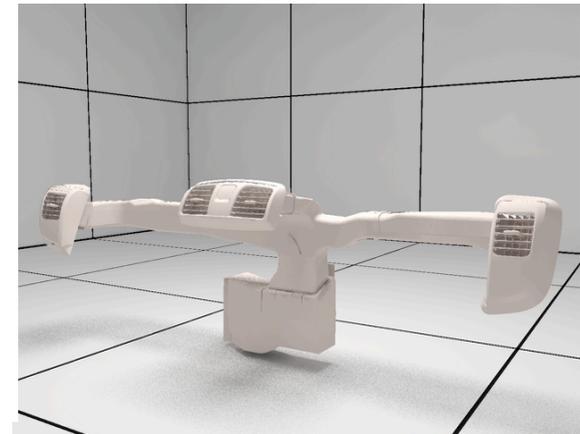
- Photo realistic rendering
- Global illumination

## ■ マルチコアレンダリング

- CPUベースの並列アルゴリズム

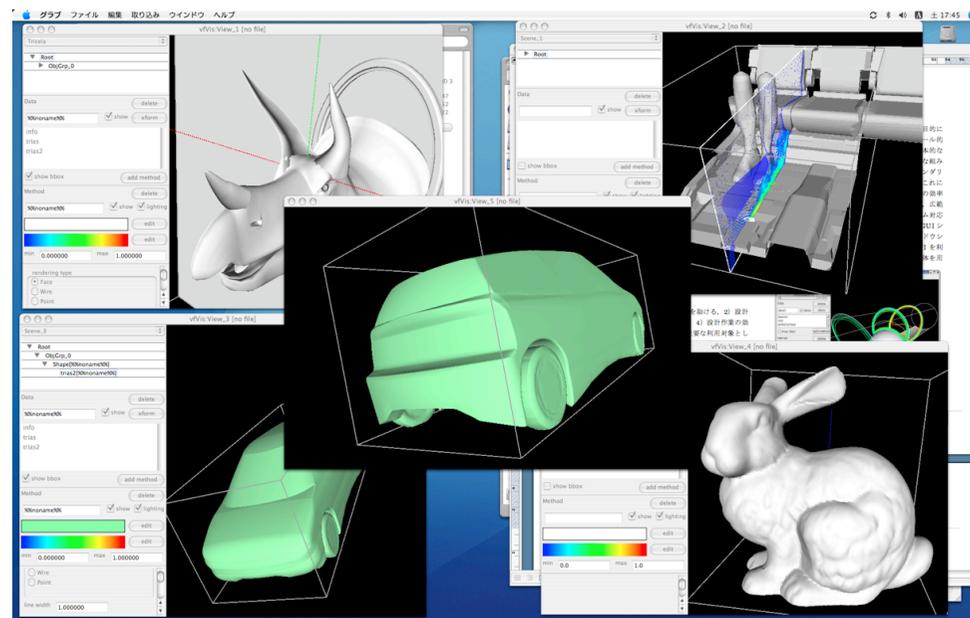
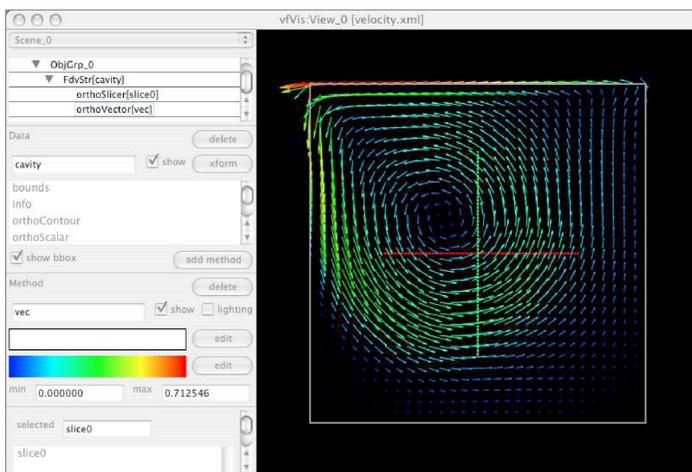
## ■ 並列GPU

- GPUを使った並列処理, イメージ 合成



# マルチプラットフォーム対応クライアント

- ■ Mac, Linux, Win対応クライアント
  - マルチプラットフォーム対応GUI
- ■ クライアントは独立アプリとして動作
- ■ サーバとクライアントの可視化機能・UIは共通
- ■ 既存のクライアントも利用可能



# インタラクティブ機能

## ■ 高速画像生成

- マルチコアレンダリング
- GPUレンダリング

## ■ イメージ重畳

- Direct Send, Binary-Swap, SLIC

## ■ 性能予測

- ネットワーク大域, レイテンシ
- 各プロセスの所要時間

## ■ データ配信

- イメージパケット
- ストリーミング

## ■ LOD技術

## ■ オブジェクト送信

## ■ ローカル/リモートのハイブリッド方式

# データ分析機能

- 統計分析
- 時間変動解析
  - FFT, Wavelet
- 乱流解析用モジュール
- 固有値解析

## *Time-Varying*データハンドリング機能

### ■ Out of Coreデータの効率的な処理

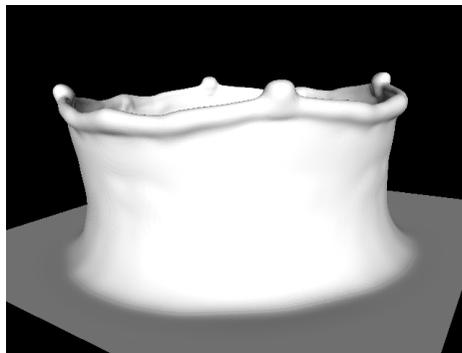
- プリフェッチ
- 並列IO
- データベース
- LODを利用したインデクシング
- ✓ Global Static Indexing for Real-time Exploration of Very Large Regular Grids, SC2001, V. Pascucci(UCD, LLNL)

### ■ データ削減技術

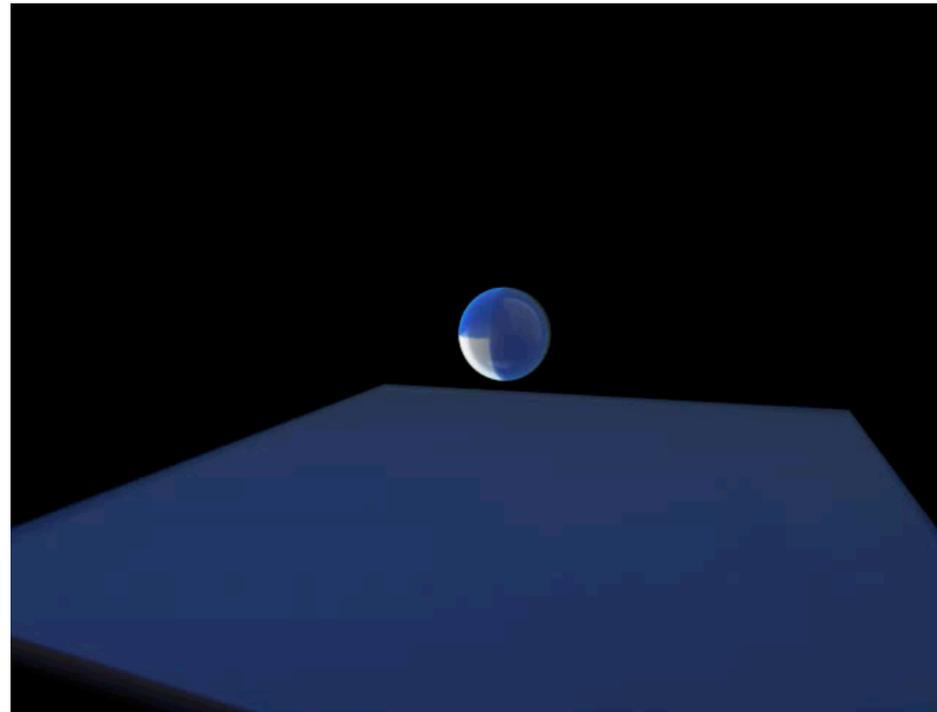
## 物理現象の可視化



写真

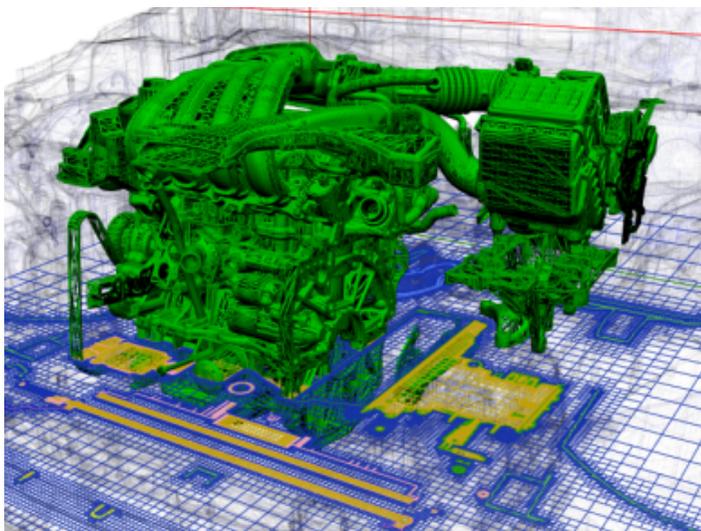


サーフェイスによる可視化



ミルククラウンのシミュレーション結果のボリューム  
レンダリングによる可視化

# 可視化のその前に！ 大規模データを作成できるか？

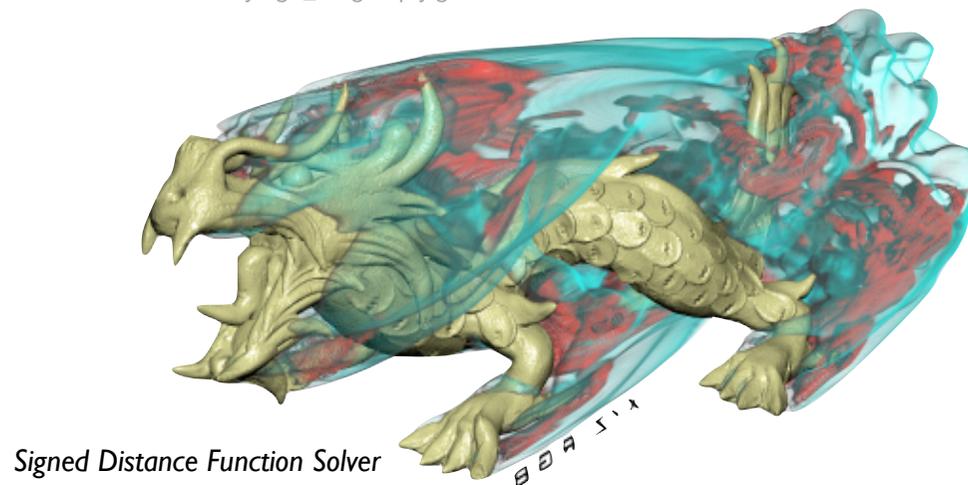


Data: Courtesy of Nissan Motor, 3.8M tria.

- Octree level: up to 16
- ✓  $4k^3$  regular cells = 64G
- ボクセル生成時間
- 221 sec.
- 大規模ボクセルの高速生成手法
- 世界最高水準
- 実用化の問題を把握. 対応



[http://graphics.stanford.edu/data/3Dscanrep/xyzrgb/xyzrgb\\_dragon.ply.gz](http://graphics.stanford.edu/data/3Dscanrep/xyzrgb/xyzrgb_dragon.ply.gz)



Signed Distance Function Solver

## まとめ

- 次世代計算機で生み出される大規模・大量のデータを処理するしくみ
- 開発システムの概要とスケジュール
- 工学利用のためには、ポスト以前にモデル作成部分が重要