RICC
RIKEN Integrated Cluster of Clusters

RIKEN

スーパーコンピュータRICCの現状と取組みについて

理化学研究所
情報基盤センター

野田　茂穂

RICCの紹介と運用状況

RICC設計コンセプト

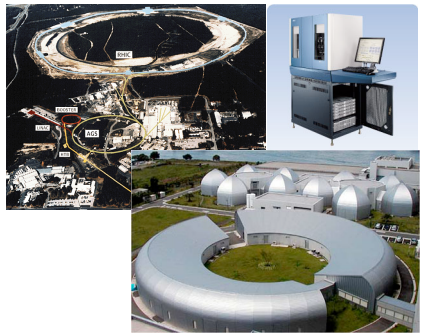
I. 次世代スーパーコンピュータを活用するアプリケーションの開発環境

II.新しい計算機技術の利用

III.加速器・次世代シーケンサなどの実験データのストレージ・解析

数千並列規模の並列ジョブ実行が可能

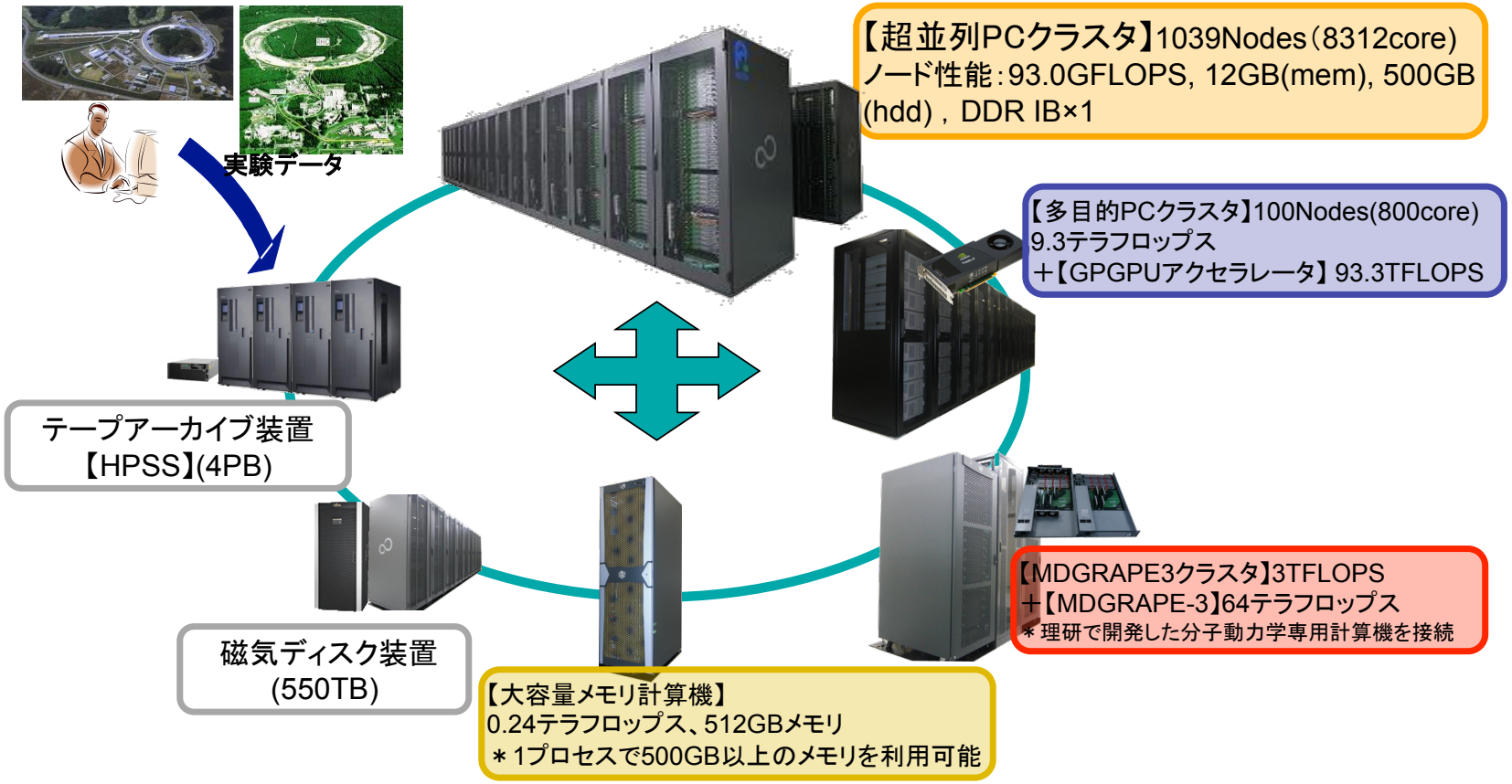
１００ノードにグラフィックボードを利用したアクセラレータを搭載



4PBのテープアーカイブ装置と500TBのディスク装置が実験結果の大規模データに対応

RICCの概要

【システム構成】
超並列PCクラスタ＋GPUクラスタ＋専用機クラスタ＋大容量メモリ計算機



各コンポーネントの特長 (利用手引書(簡易版): P2～3)

✺計算ノード

- 超並列PCクラスタ　MPC (8000->8192)
  - ➡大規模並列計算用途
- 多目的PCクラスタ　UPC (800)
  - ➡商用・フリーアプリケーション実行
  - ➡GPUアクセラレータを搭載
- インタラクティブジョブクラスタ (32)
- 占有利用課題クラスタ (160)
- MDGRAPE-3クラスタ　MDG
  - ➡分子動力学専用計算機
- 大容量メモリ計算機　AX
  - ➡1プロセスで大量メモリを必要とする計算向け

✺フロントエンドシステム

- 磁気ディスク装置
  - ➡ホーム領域(/home) : 30TB (1ユーザあたり最大500GB)
  - ➡データ領域(/data)　: 300TB
- テープアーカイブ装置(２PB)

インストールされているアプリ／ライブラリ

アプリケーション一覧

Name	ricc	mpc	upc	mdg	ax	Install directory
ADF			○			/usr/local/adf2008.01
ADF			○			/usr/local/adf2009.01
AMBER 8				○		/usr/local/amber8
AMBER 10			○			/usr/local/amber10
AMBER 11			○			/usr/local/amber11
ANSYS	○		○			/usr/local/ansys
DALTON		○	○			/usr/local/dalton
DMol3		○				/usr/local/Accelrys
GAMESS		○	○			/usr/local/gamess
Gaussian03			○			/usr/local/gaussian/g03_E01
Gaussian03NBO			○			/usr/local/gaussian_nbo/g03_E01
Gaussian09			○			/usr/local/gaussian/g09.A.02
GaussView	○					/usr/local/gaussian/gv
GROMACS		○	○			/usr/local/gromacs-4.0.5
GRRM			○			/usr/local/GRRM-1.[illegible]028
molden	○					/usr/local/molden-4.7
meep		○	○			/usr/local/meep-1.1
NAMD		○	○			/usr/local/namd-2.7b1
VMD	○					/usr/local/vmd-1.8.7
Hermes		○				/usr/local/GOLD_Suite
ROOT	○	○	○			/usr/local/root-5.26-00a
VisIt	○					/usr/local/VisIt
Smoldyn		○	○			/usr/local/smoldyn-2.12

ライブラリ一覧

Name	ricc	mpc	upc	mdg	ax	Install Directory
IMSL	○	○	○			/opt/vni/imsl/fnl600/lnxin100e64
IMSL	○	○	○			/opt/vni/imsl/fnl600/lnxin110e64
GSL	○	○	○	○		/usr/local/lib
GSL	○	○	○	○	○	/usr/local/lib
GSL	○	○	○	○		/usr/local/lib
FFTW3	○	○	○	○	○	/usr/local/lib
FFTW2	○	○	○	○	○	/usr/local/lib
FFTW2-MPI	○	○	○	○	○	/usr/local/lib
HDF5	○	○	○	○	○	/usr/local/hdf5-1.8
ATLAS	○	○	○	○		/usr/local/atlas3.8.3
Harminv		○	○			/usr/local/meep-1.1
libctl		○	○			/usr/local/meep-1.1
Xerces-C++	○	○	○	○		/usr/local/
libxml2	○	○	○	○		/usr/local/

RICCポータル

https://ricc.riken.jp

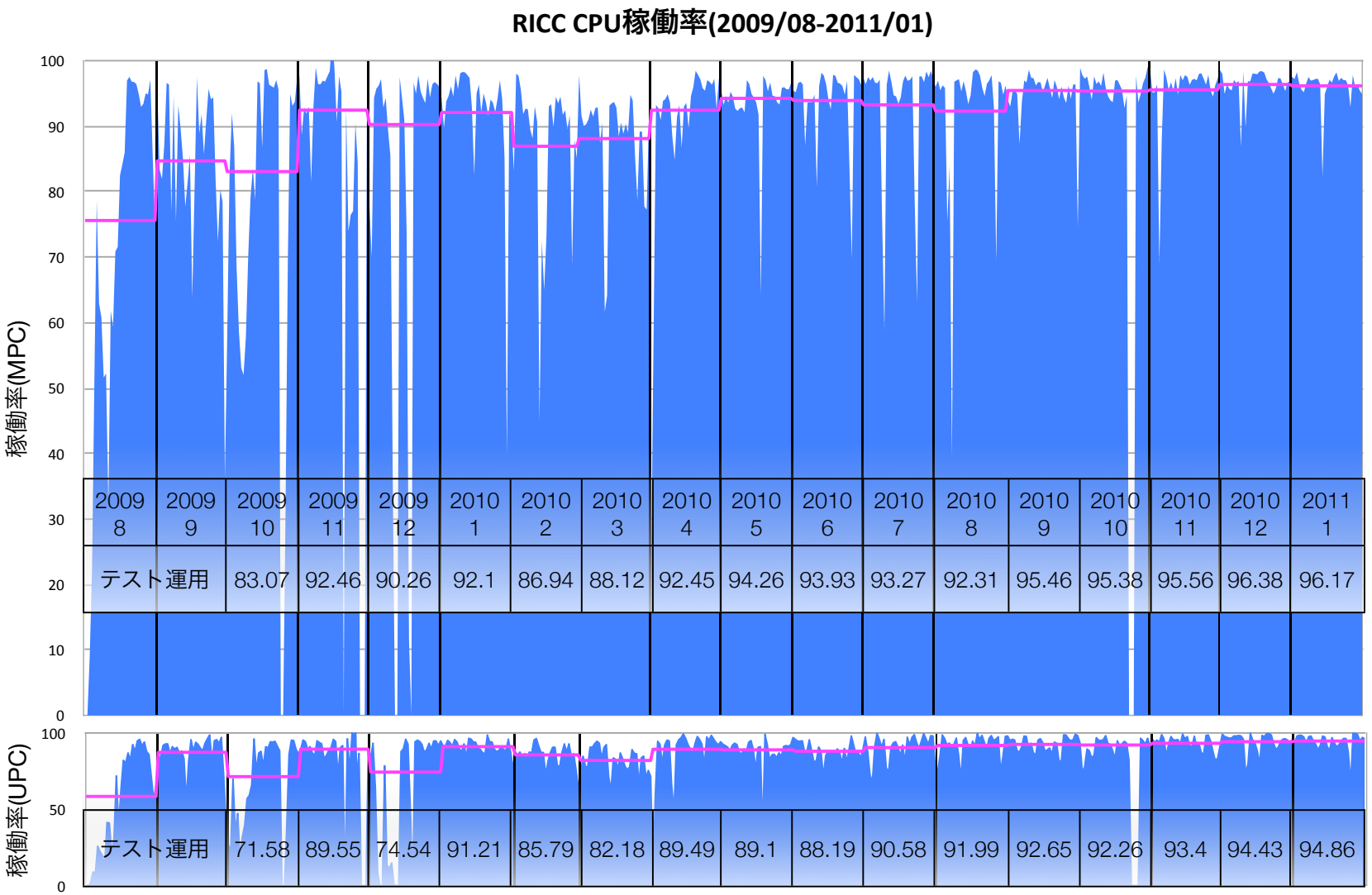
Python-2.4 ライブラリ

Name	ricc	mpc	upc	mdg	ax	Install Directory
nose	○	○	○	○		/usr/lib64/python2.4/site-packages
numpy	○	○	○	○		/usr/lib64/python2.4/site-packages
scipy	○	○	○	○		/usr/lib64/python2.4/site-packages
Rodis		○	○	○		/usr/lib64/python2.4/site-packages
mpi4py	○	○	○	○		/usr/lib64/python2.4/site-packages
Matplotlib (PyLab)	○	○	○	○		/usr/lib64/python2.4/site-packages
pytz	○	○	○	○		/usr/lib64/python2.4/site-packages
dateutil	○	○	○	○		/usr/lib64/python2.4/site-packages

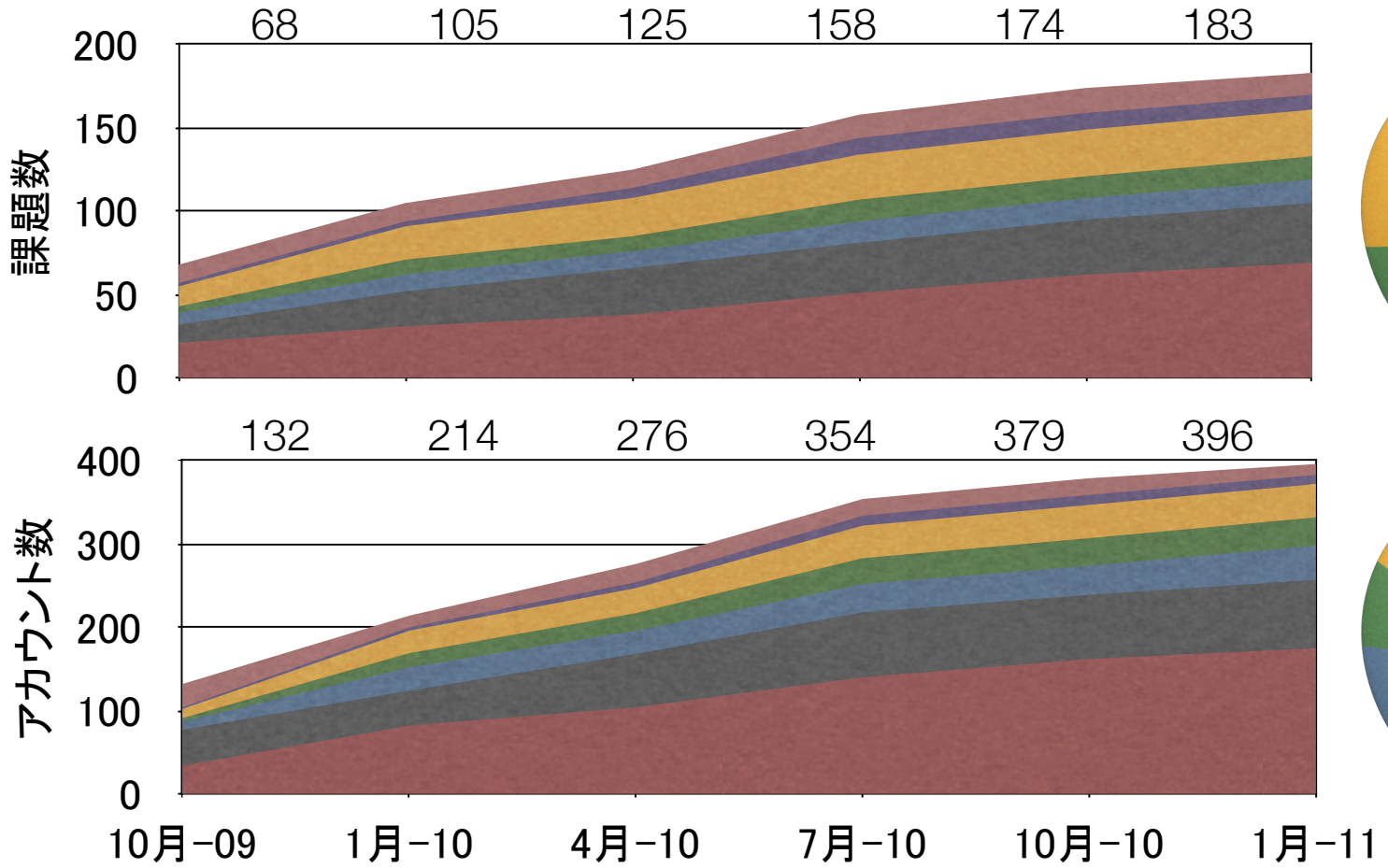
Python-2.6.5 ライブラリ

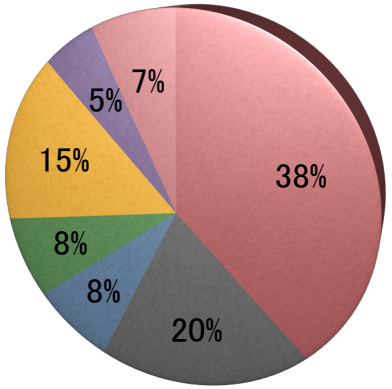
Name	ricc	mpc	upc	mdg	ax	Install Directory
nose	○	○	○	○		/usr/local/python-2.6.5/lib/python2.6/site-
numpy	○	○	○	○		/usr/local/python-2.6.5/lib/python2.6/site-
scipy	○	○	○	○		/usr/local/python-2.6.5/lib/python2.6/site-
mpi4py		○	○			/usr/local/python-2.6.5/lib/python2.6/site-

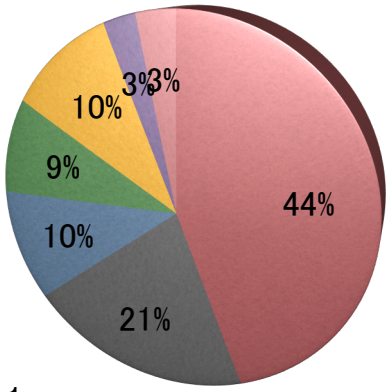
稼働率



課題数　アカウント数

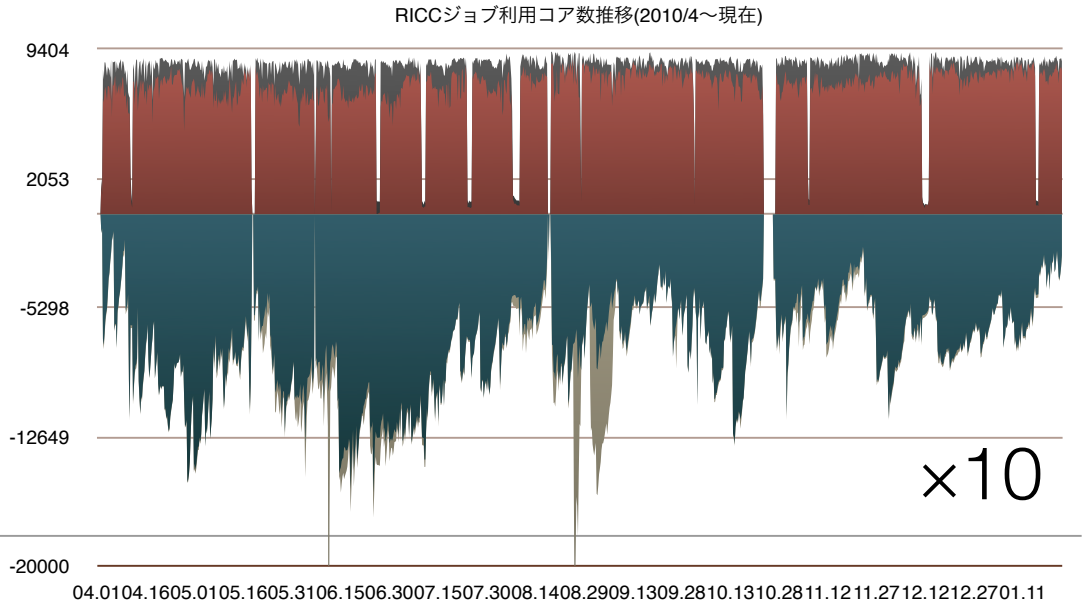






ライフサイエンス　原子核・高エネルギー物理学　化学　工学
物理学　天文学　情報工学

運用状況まとめ



PCクラスタが95%を超える稼働率

→常時約50000コア程度の待ちジョブがある

課題審査の結果を元にジョブ実行を制御

→独自に開発したMJS(メタジョブスケジューラ)

→審査で高い評価が得られている課題は希望通りの演算が実行できるように制御

高並列のジョブは週末運用（※後述）を用意

情報基盤センター取り組みについて

RICC設計コンセプト

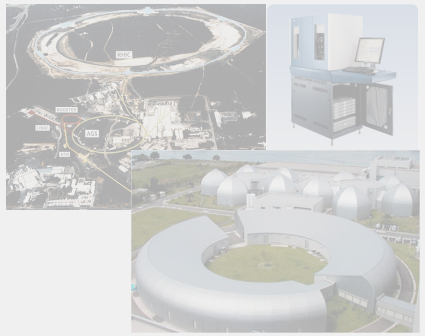
I. 次世代スーパーコンピュータを活用するアプリケーションの開発環境

数千並列規模の並列ジョブ実行が可能

II.新しい計算機技術の利用

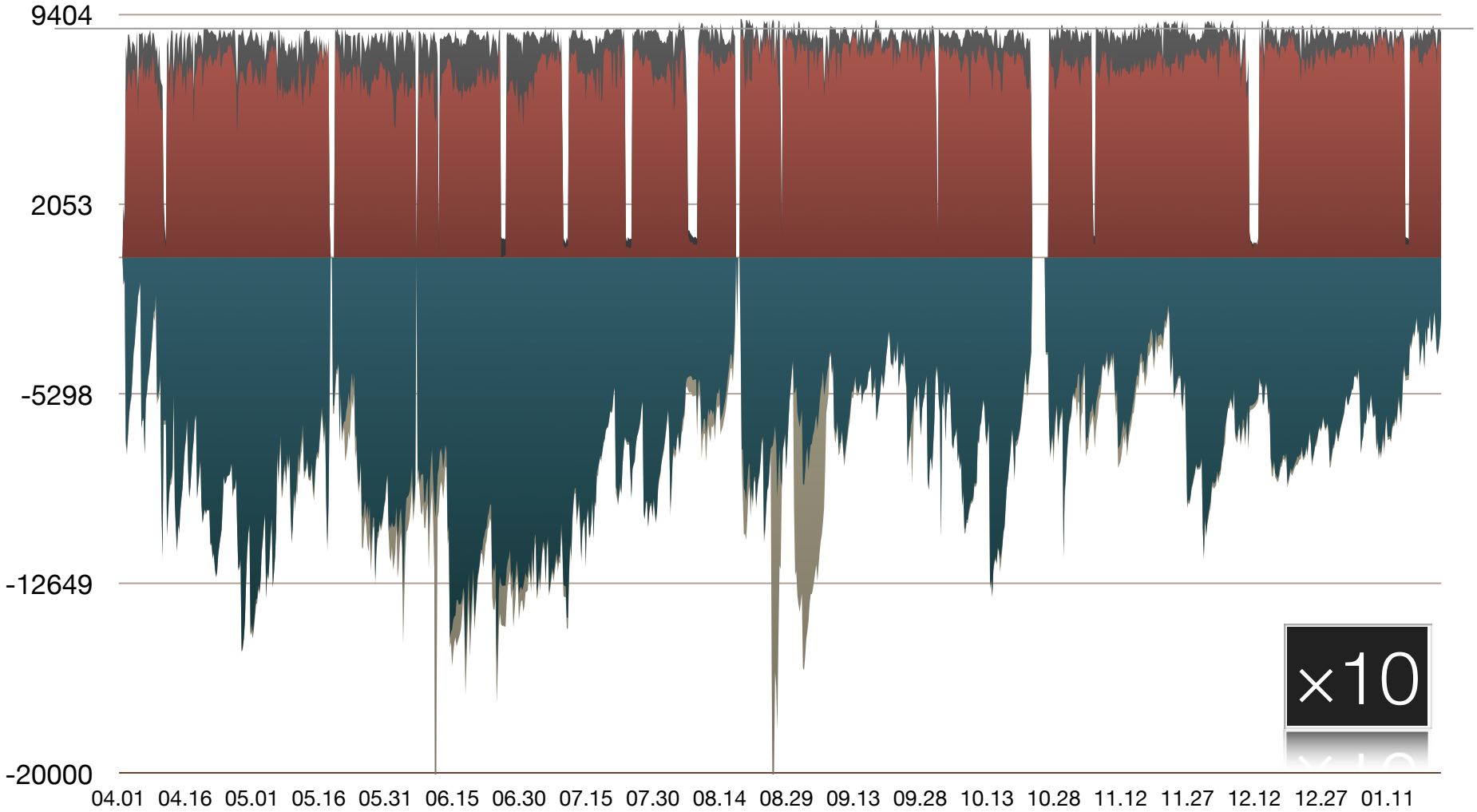
１００ノードにグラフィックボードを利用したアクセラレータを搭載

III.加速器・次世代シーケンサなどの実験データのストレージ・解析



4PBのテープアーカイブ装置と500TBのディスク装置が実験結果の大規模データに対応

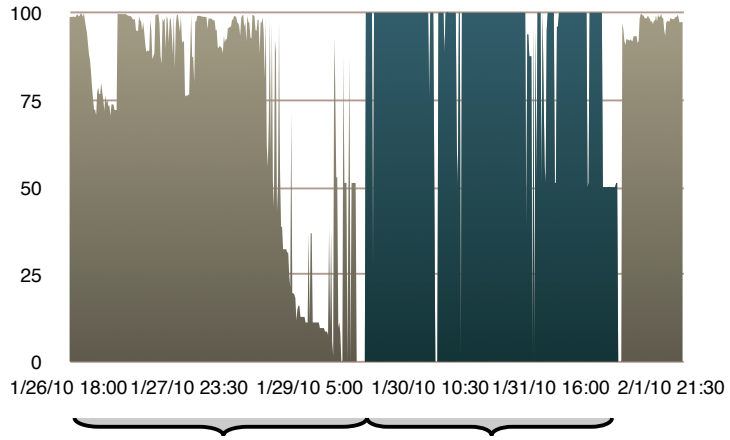
実行と待ちJobの推移

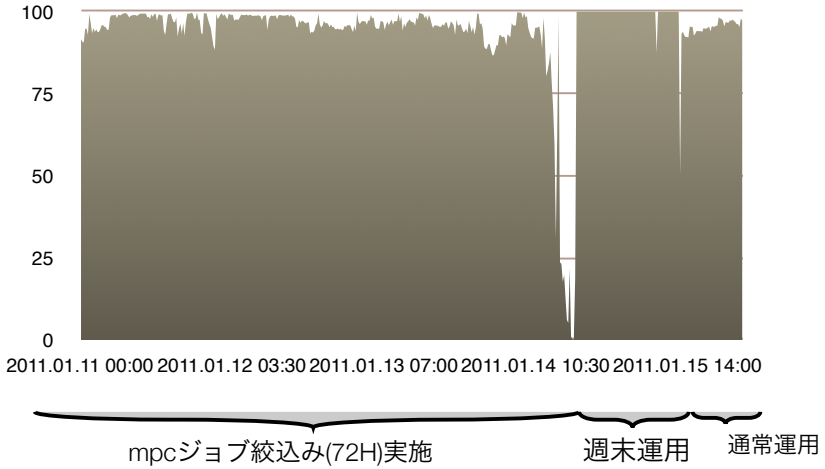


- PCクラスタが95%を超える稼働率
- →常時約50000コア程度の待ちジョブがある

大規模並列ジョブのための週末運用

- 常時高並列ジョブがサブミットできるが高い稼働率のため現実的に通常運用中の実行は困難
  - 大規模並列ジョブをまとめて実行する日を設ける
- 利用区分に関わらずどの課題でも参加できる
  - 目的：高並列なプログラム開発やチューニング
- ２０１０年１月から開始（１～２回/月のペース）





参加状況

2010年度は12回実施

参加課題数：47件（13課題）（1月まで）

各回の平均演算時間　179,560コア時間（1月まで）

→平均すると30時間程度

主な参加目的

・スケーラビリティの調査

・プログラムの高並列実行時の動作確認

参加課題の例

- タンパク質・核酸など生体高分子の分子シミュレーション（木寺 詔紀、松永　康佑、森次圭、他）
  - タンパク質の構造変化パスをサンプリングするプログラムのベンチマーク
- 大規模遺伝子ネットワーク推定プログラムの研究開発（宮野悟、玉田 嘉紀）
  - マイクロアレイデータなどの遺伝子発現データから遺伝子間の 発現制御ネットワークを推定・予測するプログラム.
- 生体高分子生化学的機能解析のための分子計算技術の開発（木寺 詔紀、中村春木、他）
  - 開発中の量子化学計算プログラム Platypus-QM のRDFT計算機能の並列性能測定
  - 異なるスケールモデル（今回は生体高分子の全原子モデルとより粗視化した残基レベルモデル）の分子動力学シミュレーションを連成した計算（MSES法）

RICC設計コンセプト

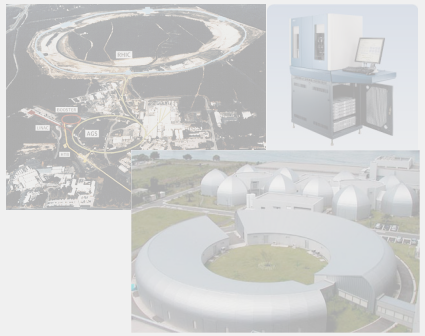
I. 次世代スーパーコンピュータを活用するアプリケーションの開発環境

数千並列規模の並列ジョブ実行が可能

II.新しい計算機技術の利用

１００ノードにグラフィックボードを利用したアクセラレータを搭載

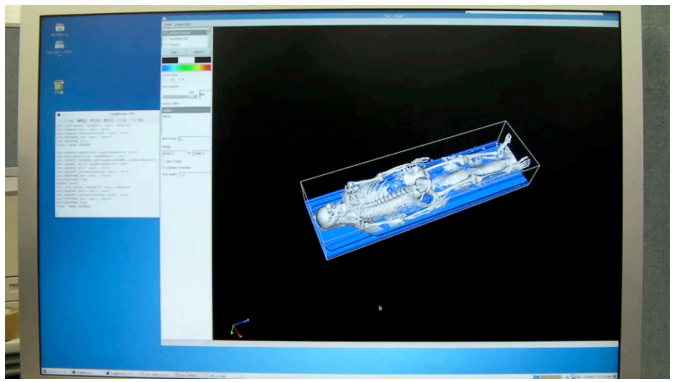
III.加速器・次世代シーケンサなどの実験データのストレージ・解析



4PBのテープアーカイブ装置と500TBのディスク装置が実験結果の大規模データに対応

アクセラレータの利用状況と利用推進

- 状況
  - アクセラレータで10倍程度の加速が可能
  - 既にプログラム移植が済んでいる研究
    - 第一原理計算による機能性物質としての分子結晶の理論解析
    - 分子動力学計算商用アプリ（Amber）
  - アクセラレータ向けアルゴリズムの研究
    - LSV ：次世代スパコンの計算結果可視化のための大規模可視化ソフトの開発



Test LSV:107GB Voxel

- 現在はアクセラレータの稼働率は低い
- アクセラレータを活用できるアプリケーションの整備が課題
  - プログラムを容易にGPGPU化出来る環境の整備
    - PGI等のコンパイラの導入
    - GPGPUライブラリの開発と利用ツール(RIVER)の整備
    - GPGPUプログラミング教育の実施
  - プログラムのGPGPU移植作業の実施

RIVER (RIKEN IBM Visual Programing Environment)

メニーコア化・ハイブリッド化の流れ

- CPUの多コア化、および多量のコアを備えたアクセラレータの混載により、HPCのピーク性能は大幅に向上した
  - すでにハイブリッド・システムが、Top 500の上位を多数占めている
- さらなるメニーコア化、ハイブリッド化の進行が予想される

RICC

Xeon 4 **コア** + GPGPU 240**コア**

Roadrunner (IBM)

Opteron 2 **コア** + PowerXCell 9**コア**

TOP 10 Systems - 11/2010

1	Tianhe-1A - NUDT TH MPP, X5670 2.93Ghz 6C, NVIDIA GPU, FT-1000 8C
2	Jaguar - Cray XT5-HE Opteron 6-core 2.6 GHz
3	Nebulae - Dawning TC3600 Blade, Intel X5650, NVidia Tesla C2050 GPU
4	TSUBAME 2.0 - HP ProLiant SL390s G7 Xeon 6C X5670, Nvidia GPU, Linux/Windows
5	Hopper - Cray XE6 12-core 2.1 GHz
6	Tera-100 - Bull bullx super-node S6010/S6030
7	Roadrunner - BladeCenter QS22/LS21 Cluster, PowerXCell 8i 3.2 Ghz / Opteron DC 1.8 GHz, Voltaire Infiniband
8	Kraken XT5 - Cray XT5-HE Opteron 6-core 2.6 GHz
9	JUGENE - Blue Gene/P Solution
10	Cielo - Cray XE6 8-core 2.4 GHz

直面するプログラミングの困難

- メニーコア化、ハイブリッド化が進むとプログラミングが課題になる

タスク並列化

アーキテクチャごとの最適化

- 処理AはCPUで、処理BはGPUで・・・
- キャッシュサイズ、命令セットの違いを意識した最適化・・・

通信の実装

- CPU同士はMPIで、GPUとは専用の転送で・・・
- CPUの演算とGPUの演算と通信は同時に実行・・・

計算リソースへの割り当て

- 処理AはCPUでN並列、処理BはGPUだからM並列で・・・

性能と生産性を両立させる新しい枠組みが必要

ハイブリッド・システムプログラミング環境 “RIVER”

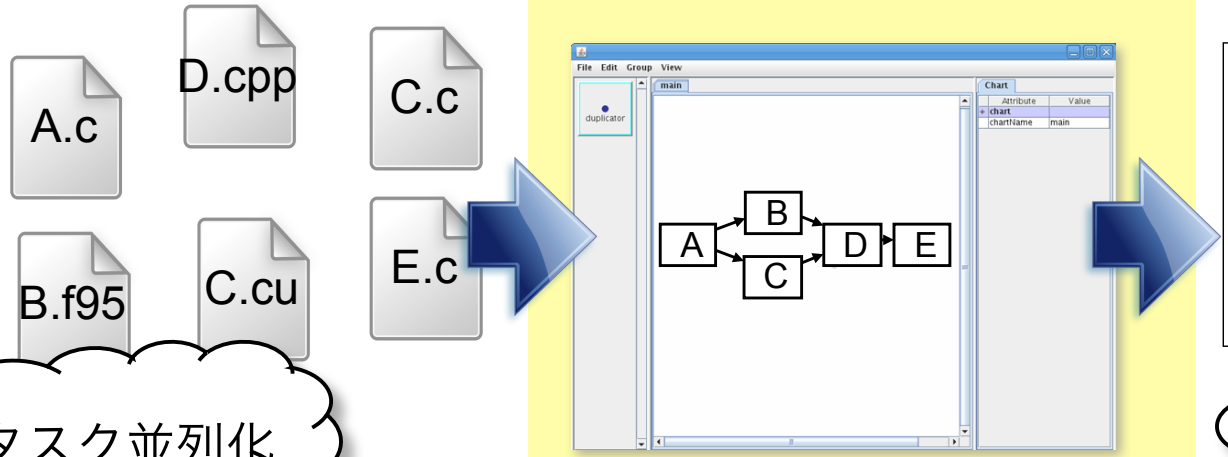
2レベルプログラミング

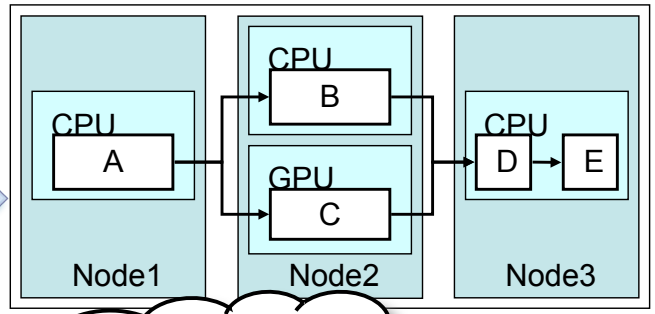
自動並列化

コンポーネント・プログラミング

データ・フロー・プログラミング

コンパイル





タスク並列化

アーキテクチャごとの最適化

通信の実装

計算リソースへの割り当て

熟練プログラマーにより作成したものを共有
シングルスレッド用に書かれた既存の言語・資産を流用可能
追加はわずかな指示文だけ

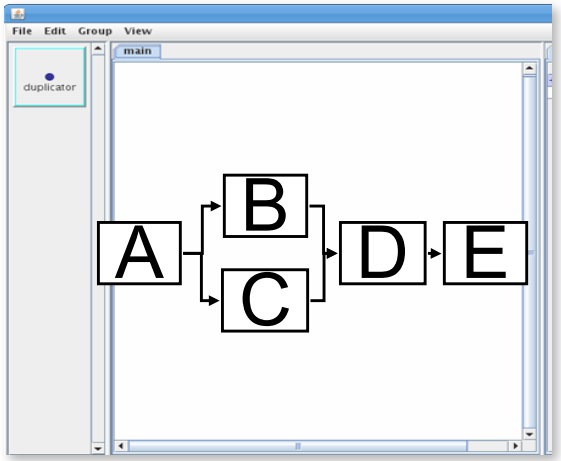
一般プログラマーは、アルゴリズムのデータフローを記述するだけ
最適化された部品ライブラリと（左）、利用するシステムに最適化された並列化（右）を自動的に利用できる

コンパイラーが、プロセッサーごとの処理能力とシステム構成を考慮して処理の割り当てと並列度を判断
通信コードも自動的に挿入

RIVERを利用したプログラミングのイメージ

1. アルゴリズムのデータフローを記述する

計算機利用者

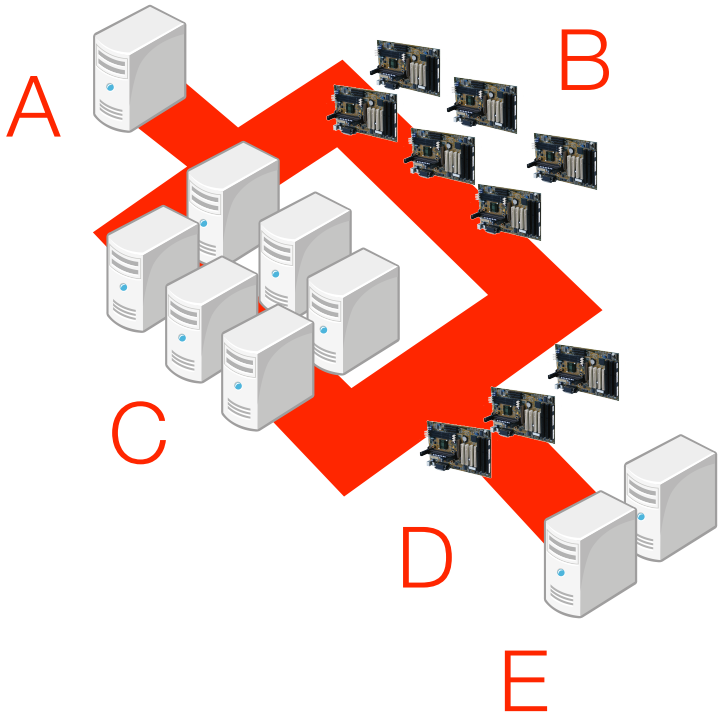


2. システム構成を与えてコンパイル

3. コンパイラーが、プログラムを生成

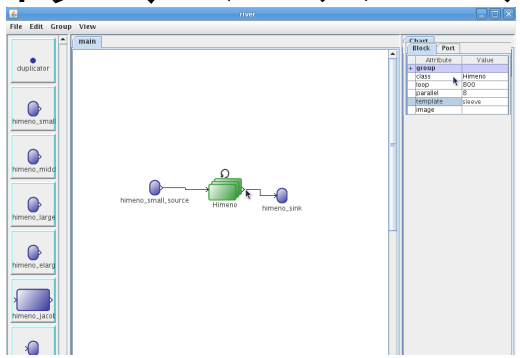
処理内容とコードライブラリを参照して、最適な計算ノード、並列数を決定。

必要なプログラム部品、通信コードを含んだ実行形式を、ノードごとに生成。(MPMD)

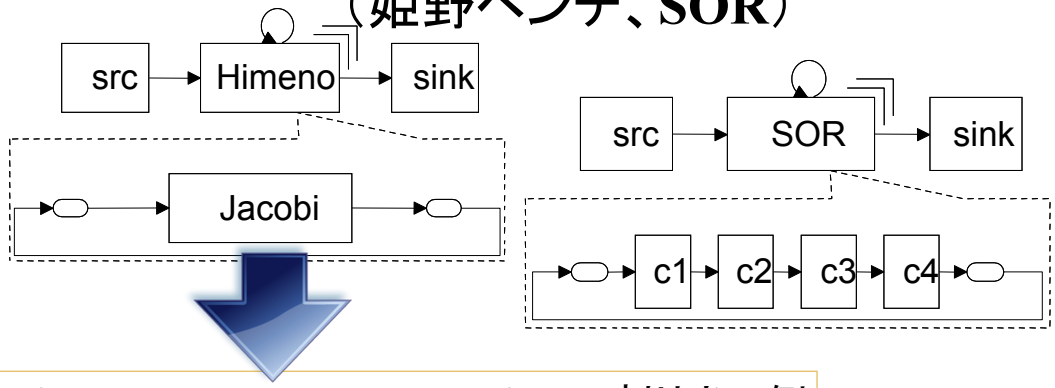


2010年度の成果

プログラミング環境の開発（コンパイラー、ビジュアル・エディタ）



部品ライブラリ開発、性能実験（姫野ベンチ、SOR）



HimenoBMT (M size = 128 x 128 x 256) での割り当て例

実験環境

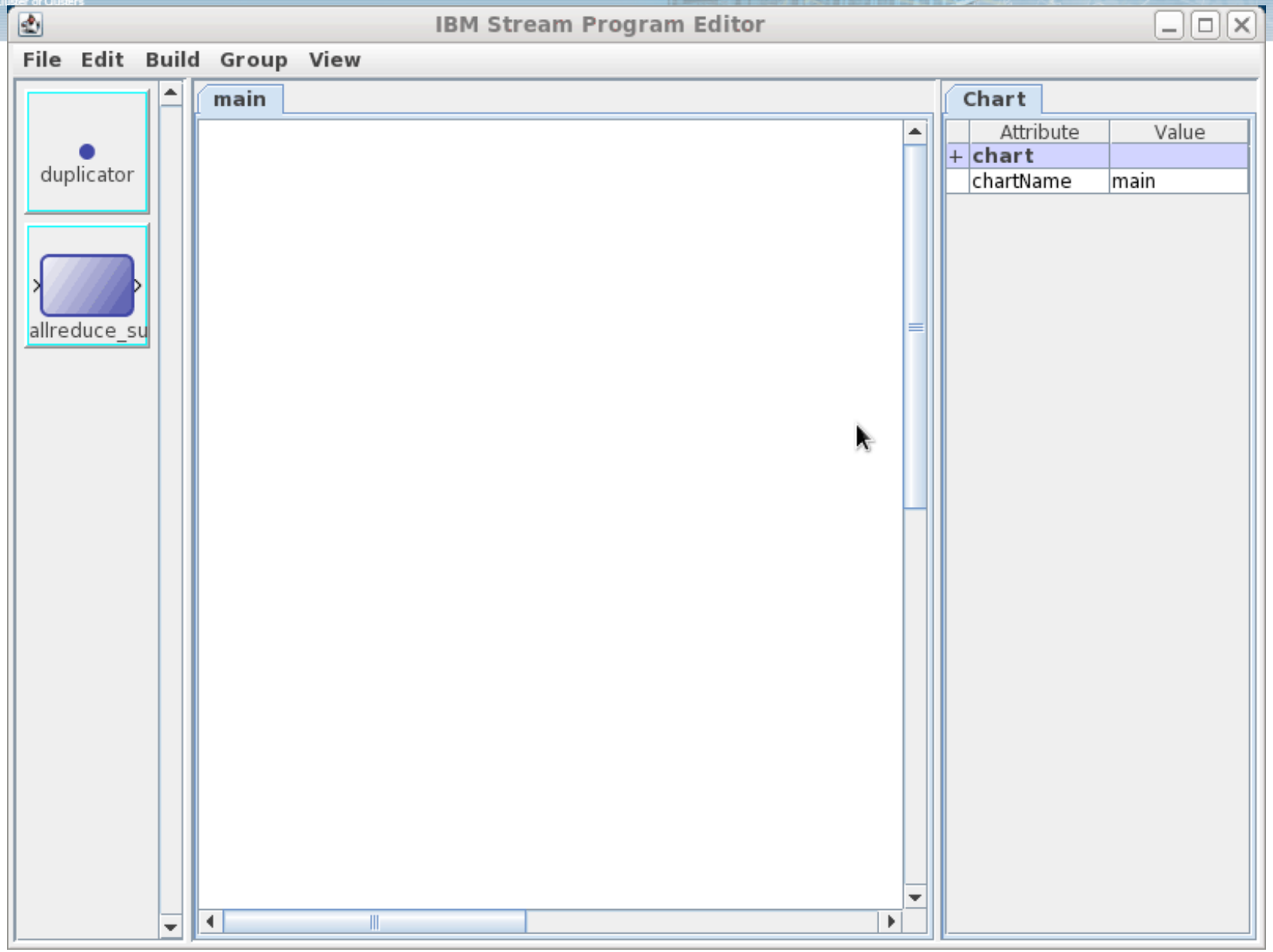
	x86	GPGPU
言語	C	C
コンパイラー	icc	nvcc
並列度	1-80	1-80
コア数またはカード数	1-80コア	1-80カード
使用ノード数	10	80
特記事項	8コア/ノー	CUDA for C

ホスト数	スケジューリング結果
1	GPGPU x 1
2	GPGPU x 1
4	GPGPU x 4
8	GPGPU x 8
16	GPGPU x 16
32	GPGPU x 16
64	GPGPU x 16
80	GPGPU x 16

2ホストは通信コストが高く1並列で実行

ホストを増やすとスケールする領域

16ホスト以上は頭うちと判断



実験結果

自動並列化・スケジューリング結果

HimenoBMT (L size = 256 x 256 x 512)

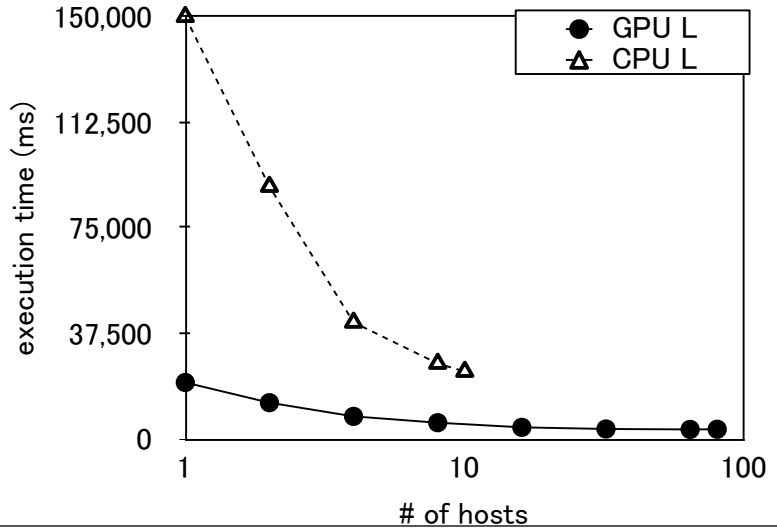
使用可能ホスト数	スケジューリング結果
1	GPGPU x 1
2	GPGPU x 2
4	GPGPU x 4
8	GPGPU x 8
16	GPGPU x 16
32	GPGPU x 32
64	GPGPU x 64
80	GPGPU x 80

スケジューリングミス；最速構成は GPGPU x 64

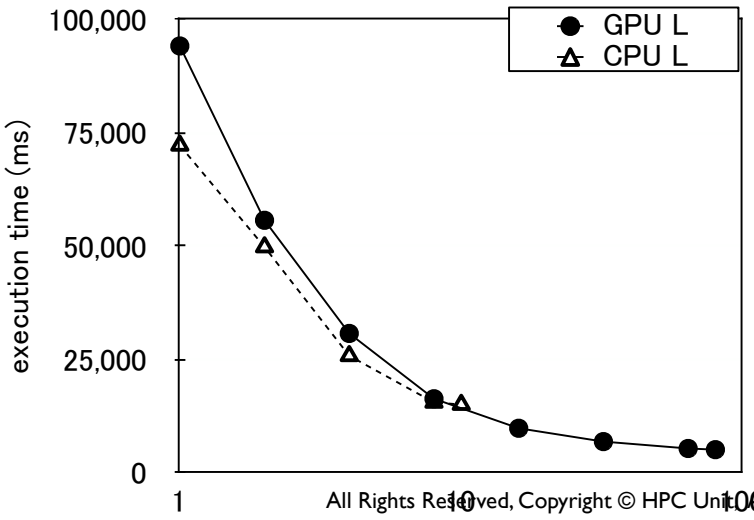
SORBMT (L size = 512 x 512 x 512)

使用可能ホスト数	スケジューリング結果
1	x86 x 8 コア
2	x86 x 16コア
4	x86 x 32コア
8	x86 x 64コア
16	GPGPU x 16
32	GPGPU x 32
64	GPGPU x 64
80	GPGPU x 80

性能特性



性能特性



ロードマップ

	2010年度	2011年度
目標	基幹ツールの開発	RICCでのパイロット運用
言語仕様	部品化指示構文の定義	
ビジュアルエディター	プロトタイプ	Eclipseプラグイン化 デバッグ支援機能
コンポーネントライブラリ	姫野ベンチ SORベンチ	実用的なアプリケーション
コンパイラー（並列化スケジューラー）	CPUおよびGPUに対応した自動スケジューリング 通信コードの自動生成（MPI） 2011/04公開予定	通信隠蔽 計算能力にあわせた自動領域分割 MPIとスレッドの通信ハイブリッド化

RICC設計コンセプト

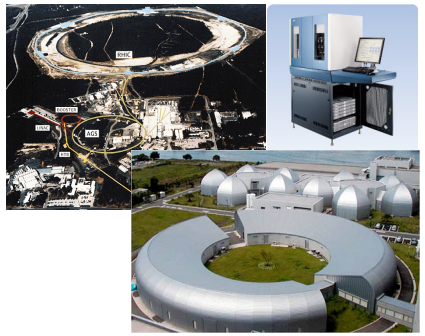
I. 次世代スーパーコンピュータを活用するアプリケーションの開発環境

数千並列規模の並列ジョブ実行が可能

II.新しい計算機技術の利用

１００ノードにグラフィックボードを利用したアクセラレータを搭載

III.加速器・次世代シーケンサなどの実験データのストレージ・解析



4PBのテープアーカイブ装置と500TBのディスク装置が実験結果の大規模データに対応

開発中：データデポジトリシステム（7月開始予定）

概要：

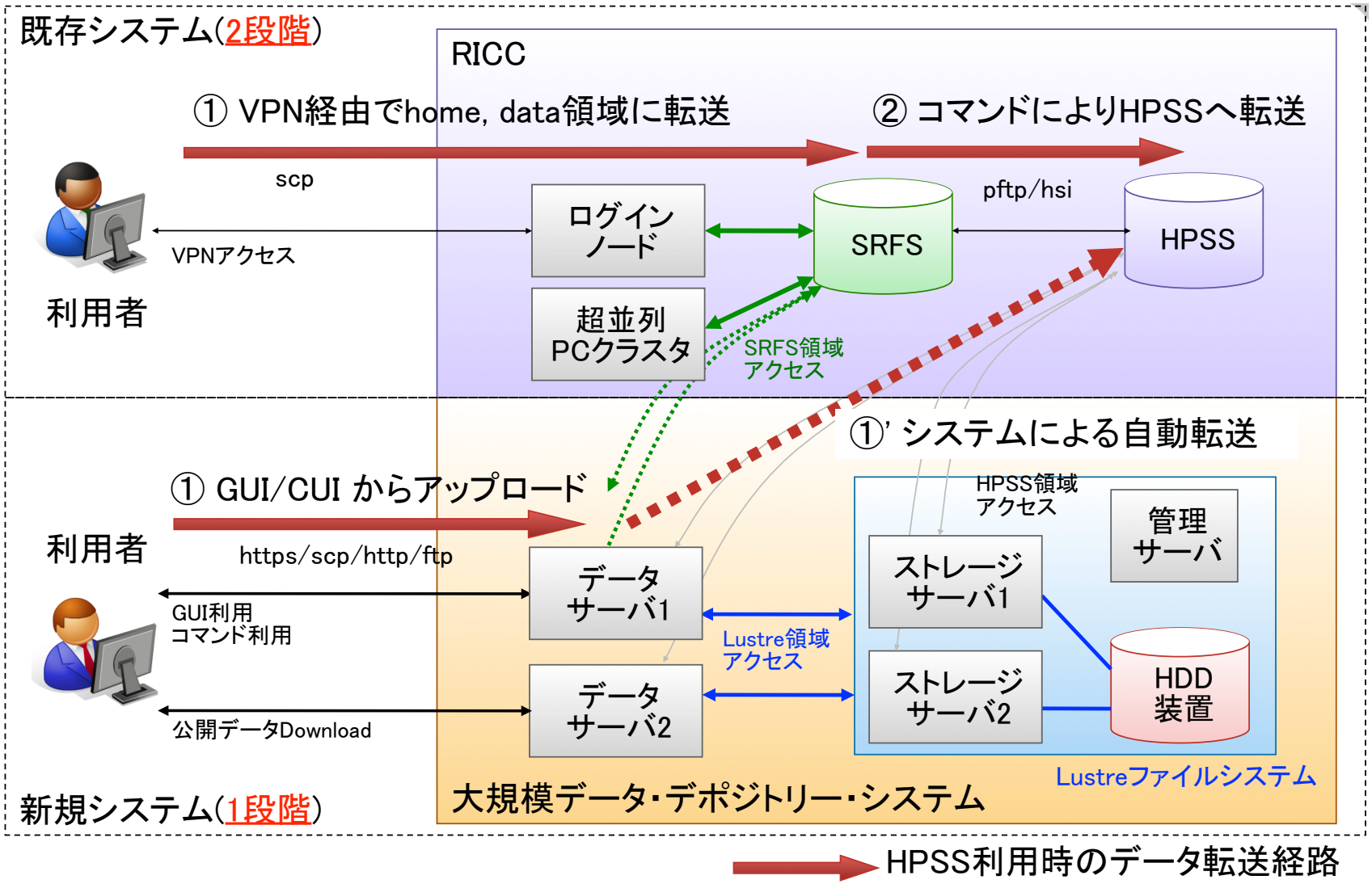
・大量の実験データが生成され、ストレージ容量が不足

→演算資源を使わないがHPSSテープアーカイブのみを利用

・HPSSを利用するには複数のコマンド実行が必要

→GUIを開発する事により、利便性の高いシステムを開発

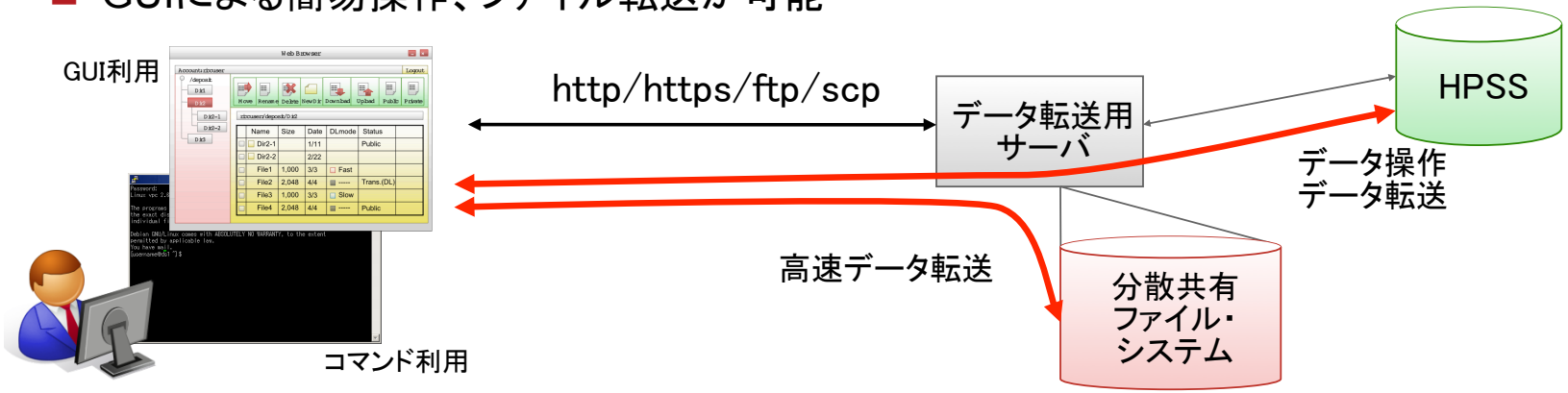
システムの利用イメージ



利用者アクセスイメージ

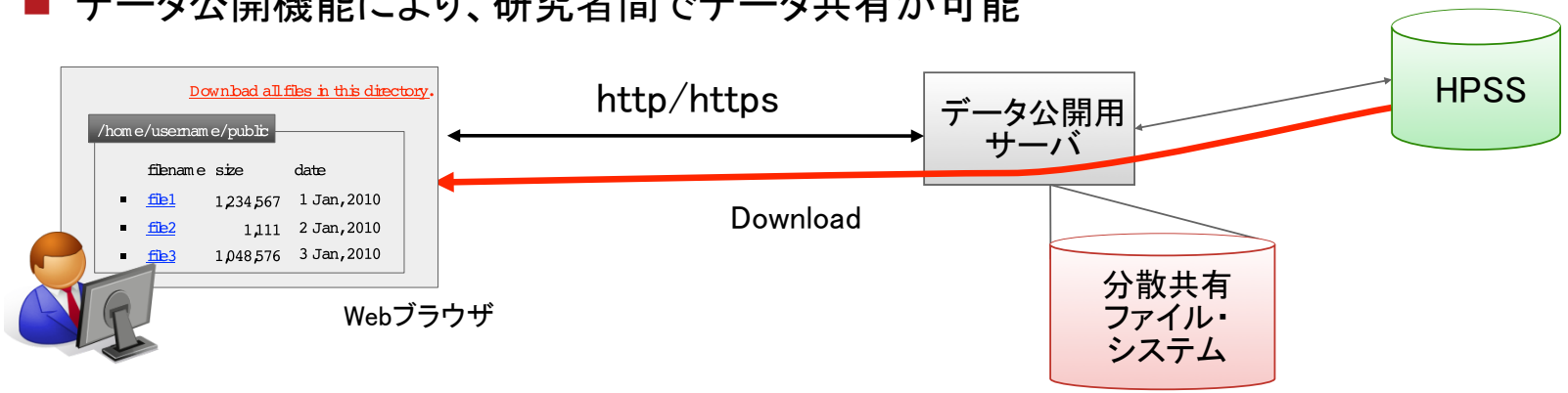
① データ操作・転送

- GUIによる簡易操作、ファイル転送が可能



② 公開データDownload

- データ公開機能により、研究者間でデータ共有が可能



その他の取り組み

チューニング

　単体性能、並列性能

講習会（RICCユーザ向けGPGPU講習会　２／１８、一般向けも計画中）

研究の状況に応じた柔軟な対応

　緊急の場合、一時的にJobの優先度をコントロール。

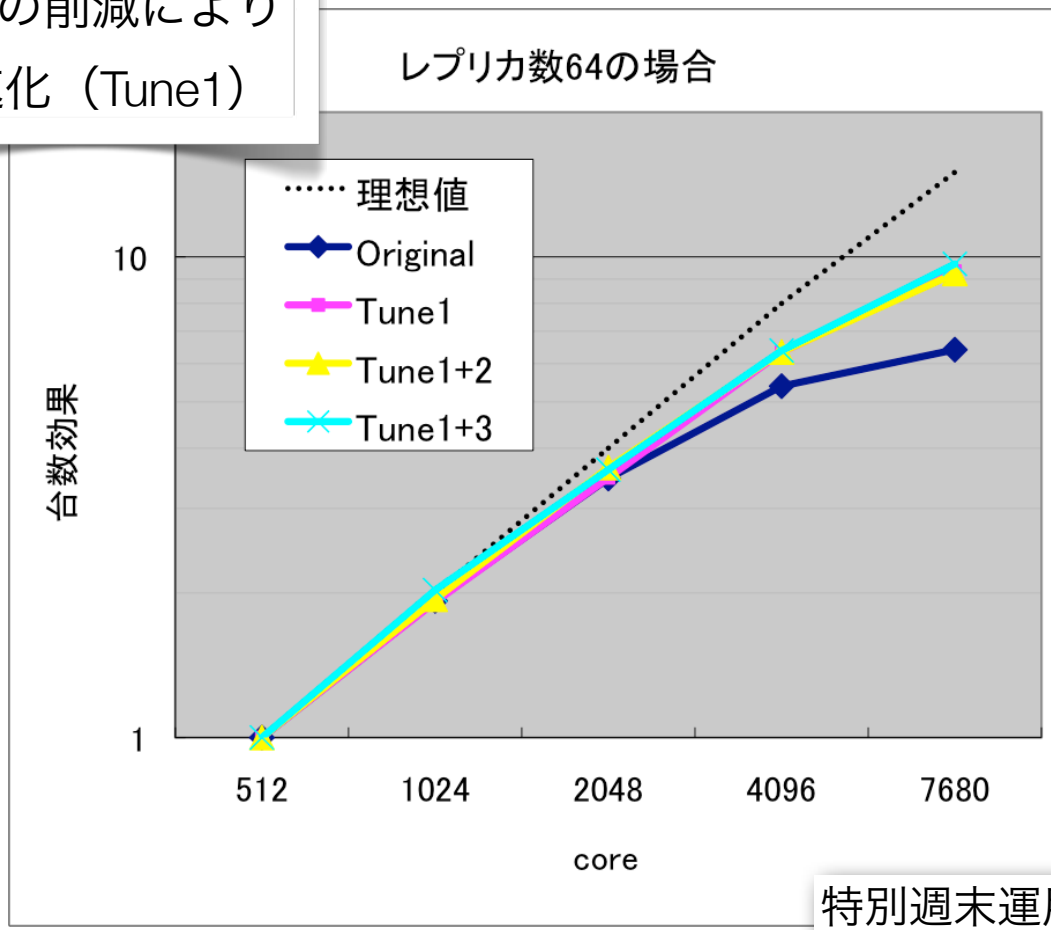
　簡易利用でも一時的に使用可能core数制限を解除

　アプリケーション／ライブラリのインストール

チューニング支援の例（粗視化モデル計算CafeMol）

ISLiM

通信の削減により
高速化（Tune1）



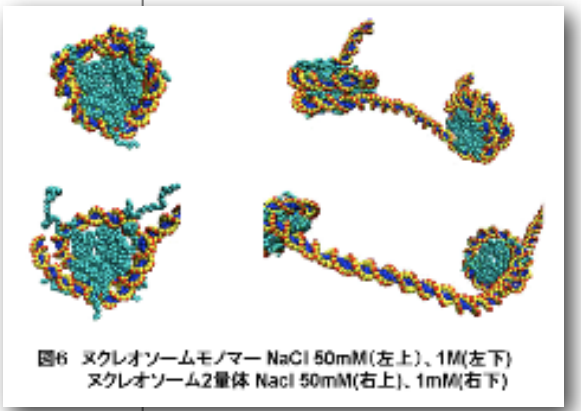


図6 ヌクレオソームモノマー NaCl 50mM(左上)、1M(左下)
ヌクレオソーム2量体 Nacl 50mM(右上)、1mM(右下)

- Tune1：通信の削減（使用しない配列要素のレプリカ内通信をやめる）
- Tune2：通信ルーチンの最適化
  ALLREDUCE→REDUCE_SCATTER +ALLGATHERV
- Tune3：通信ルーチンの最適化
  ALLREDUCE→REDEUCE+BCAST

特別週末運用にて大規模並列での性能問題を確認
情報基盤センターがチューニングを実施

まとめ

RICCは非常に高い稼働率

→より多くのジョブが実行できるよう運用に努める

→プログラムのチューニング支援により効率的な実行を支援

→アクセラレータの利用を進めて行く

ご意見、ご要望等：hpc@riken.jp