グリッドコンピューティングが拓く新しい計算科学

佐藤三久

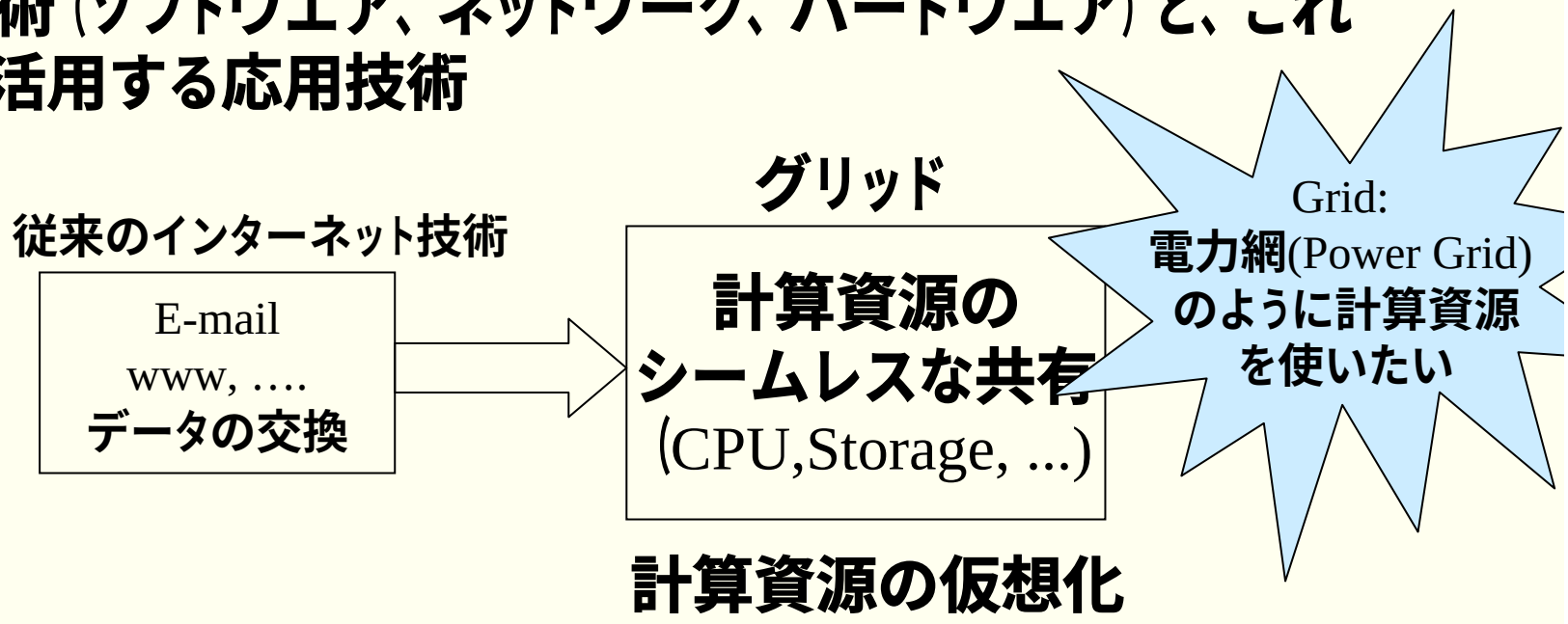
計算科学研究センター、筑波大学

もくじ

- グリッドコンピューティングとは
- 計算科学センターのグリッドコンピューティング研究の紹介
  - OmniRPC
  - CONFLEX-G
  - HMCS-G
  - ILDG
- グリッドコンピューティングと計算科学

グリッドコンピューティングとは

- グリッド技術とは広域の高速ネットワーク上において、「安全に」大量のデータ、計算資源、貴重な装置等を共有し、協調作業、資源の有効活用するネットワーク基盤技術（ソフトウエア、ネットワーク、ハードウエア）と、これを活用する応用技術



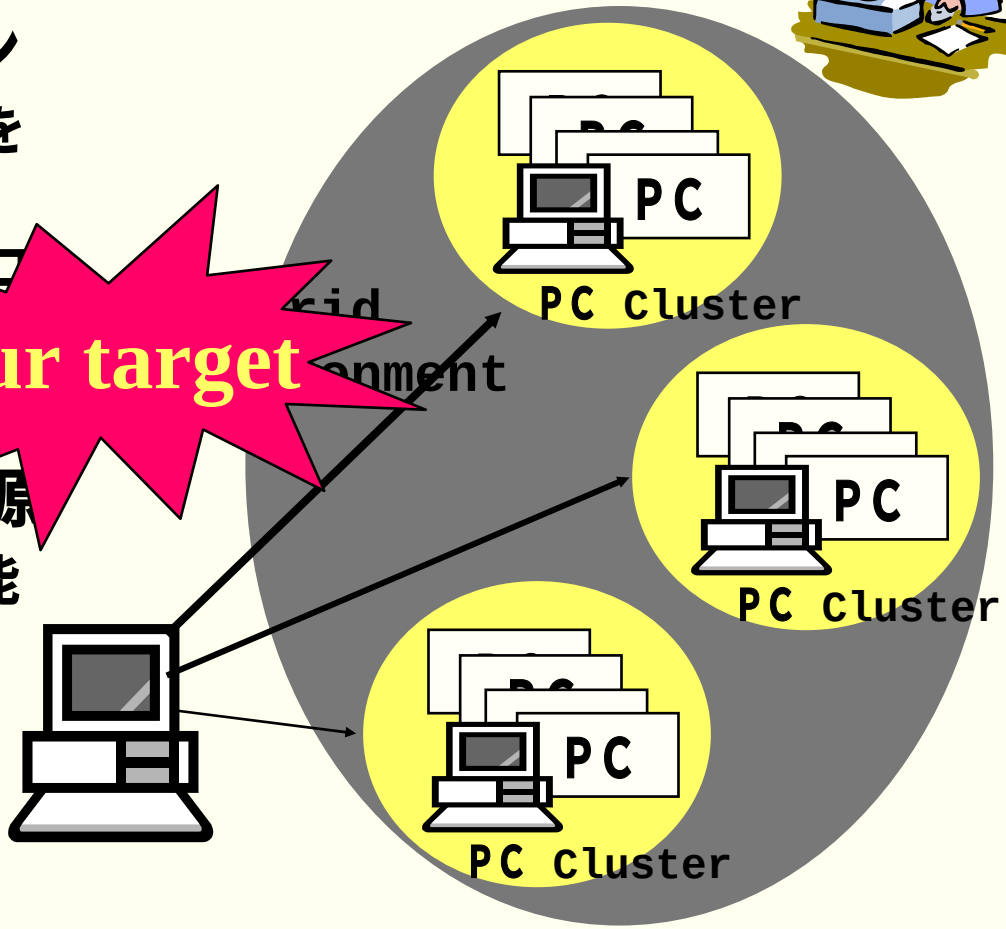
グリッドコンピューティングとは

- **なにが変わったのか？**
  - **以前から、分散コンピューティングの研究はあった。**
  - **インターネットの急激な進歩、普及**
  - **全世界で共通な基盤を作ろうとしている！**(Grid Forum, …,ApGrid)

- **何につかえるのか？**
  - **計算資源（スーパーコンピュータ）を共有し、大規模計算を行う**(meta-computing, MPI-G, …, ITBL?)
  - **遠隔の計算資源と手元の**PC/WS**をシームレスに結合**(computing portal, GridPRC)
  - **大量のデータの処理**(data intensive computing, gfarm for ATLAS)
  - **高価な装置の遠隔共有 (電子顕微鏡、衛星データ,加速器？)**
  - **共同作業のサポート(電子会議システム、**AccessGrid)
  - **遊休の計算機の利用、**SETI@home, P2P, …, XtermWeb

グリッドと並列アプリケーション

- 典型的なアプリケーション
  - パラメータ検索：同じ計算を膨大な計算資源で実行
  - master-slave型の並列プログラム
- 典型的なグリッド計算資源
  - 複数のクラスタが利用可能
  - 計算資源の状況が動的に変化

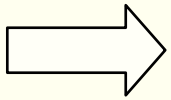


Gridの並列プログラミング

- globusのGSHベースのjobのsubmit
- MPICH-G
  - 汎用だが、すべて書かなくてはならない。
  - 資源のリソースの増減に対応できない。

```
Ninf_call_async(A,…)
Ninf_call_async(B,…)
…
Ninf_call_wait()
```

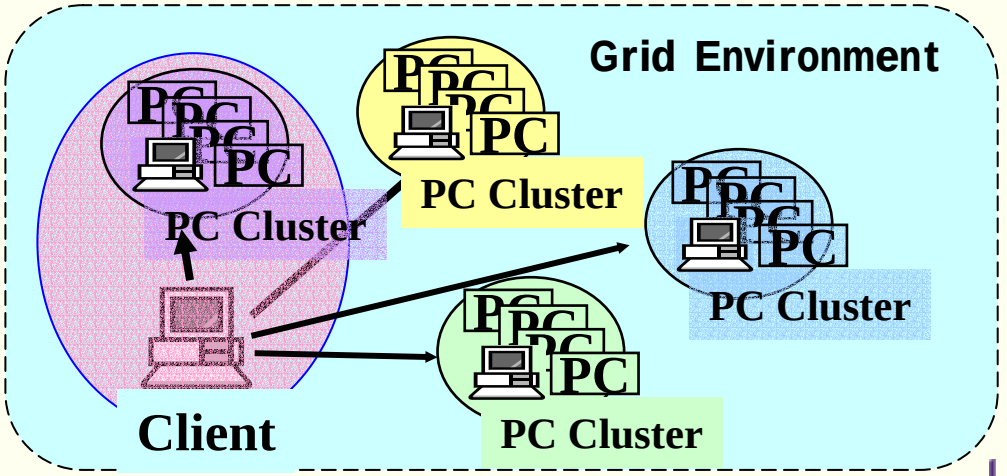
- Grid RPC: Ninf, NetSolve
  - 直感的でわかりやすい
    プログラミングインタフェース
  - 並列プログラミングは 非同期なRPC
    - activeなrequestをユーザが管理
  - Ninf ver.1では認証、セキュリティがサポートされていなかった
    - Ninf-G ではGlobusを用いているが, 起動に時間がかかる
  - private IP addressのクラスタのサポートが無い
  - ファイヤウォール内の計算資源へのサポートがない
  - ….



OmniRPC

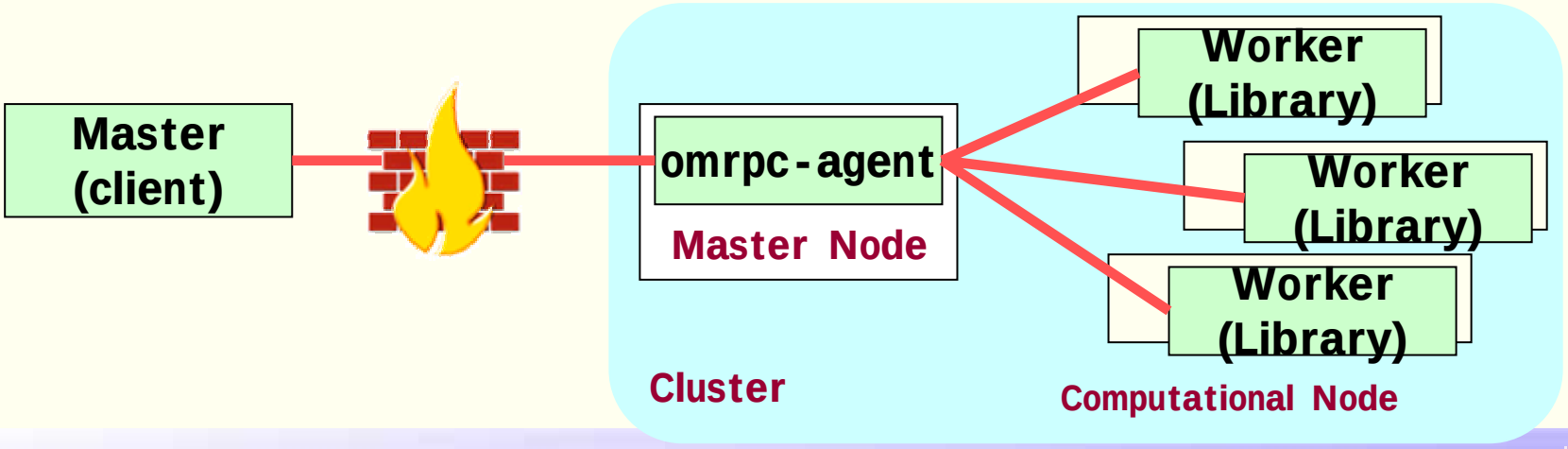
OmniRPC

- **ローカルなクラスタからグリッドでのマルチクラスタまでシームレスなプログラミング環境を提供**
  - **ローカルな環境でプログラム開発、グリッドで大規模実行**
- **主に**Master/Worker**の並列プログラムを対象**
  - **パラメータサーチアプリケーションを効率よく処理できるようにサポート**
- **簡便な並列プログラミングインタフェース**
  - Ninf**の**API**をベースにした** thread-safe**な**RPC
  - **既存の (逐次) プログラムを簡単に**Grid**の並列プログラムに移行可能**
- RPC**クライアントライブラリが負荷分散機能を持つ**



OmniRPC

- Gridのend-pointの計算環境としてクラスタを対象
  - グリッド環境では認証としてGlobusのほかにsshも可能
  - プライベートアドレスのクラスタについてもサポート
- 1000worker規模までの大規模な分散環境への対応
- 簡便な環境設定：プライベートなGrid計算環境構築
  - 使える計算リソースの指定
  - リモート実行プログラムはユーザごとに管理



OmniRPCを使ったプログラミング

- B[i]=A*C[i] (i= 1,…,100)**の行列演算をRPCを使い並列に処理**

```
int main(int argc, char **argv){
        int i, A[100][100],B[100][100][100],C[100][100][100];
        OmniRpcRequest reqs[100];
        OmniRpcInit(&argc, &argv);

        /* set matrix A in worker modules*/
        for(i = 0; i< 100; i++)
                reqs[i] = OmniRpcCallAsync("mul",100, B[i], A, C[i]);
        OmniRpcWaitAll(100,reqs);
        .
        OmniRpcFinalize();
        return 0;
}
```

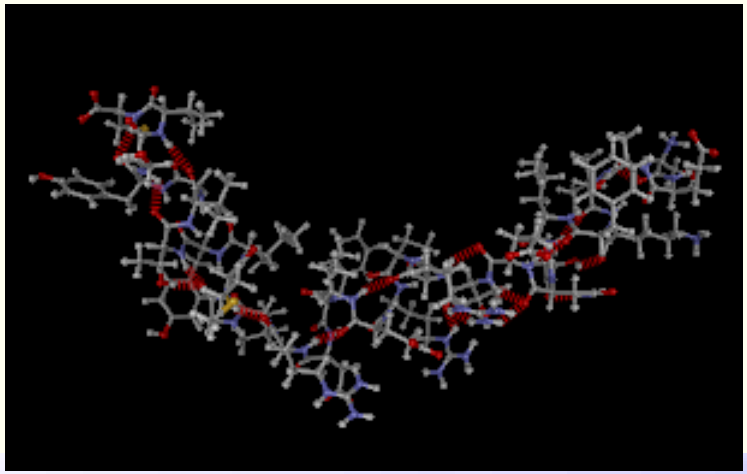
非同期でRPC呼び出しで並列化

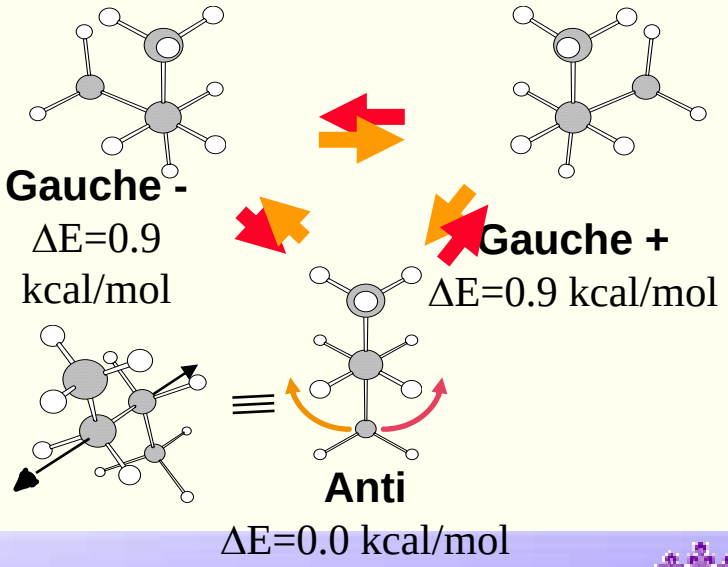
mulはB[i] = A*C[i]の
結果を返すリモート関数

RPCの終了を待つ

CONFLEX (1)

- 分子の配座(3次元構造) を網羅的に計算するプログラム
  - 豊橋技術科学大学　後藤らが開発
- 同じ分子でもエネルギー状態が異なる3次元構造をたくさん持つ
  - 大量の構造を高速に配座探索する必要性



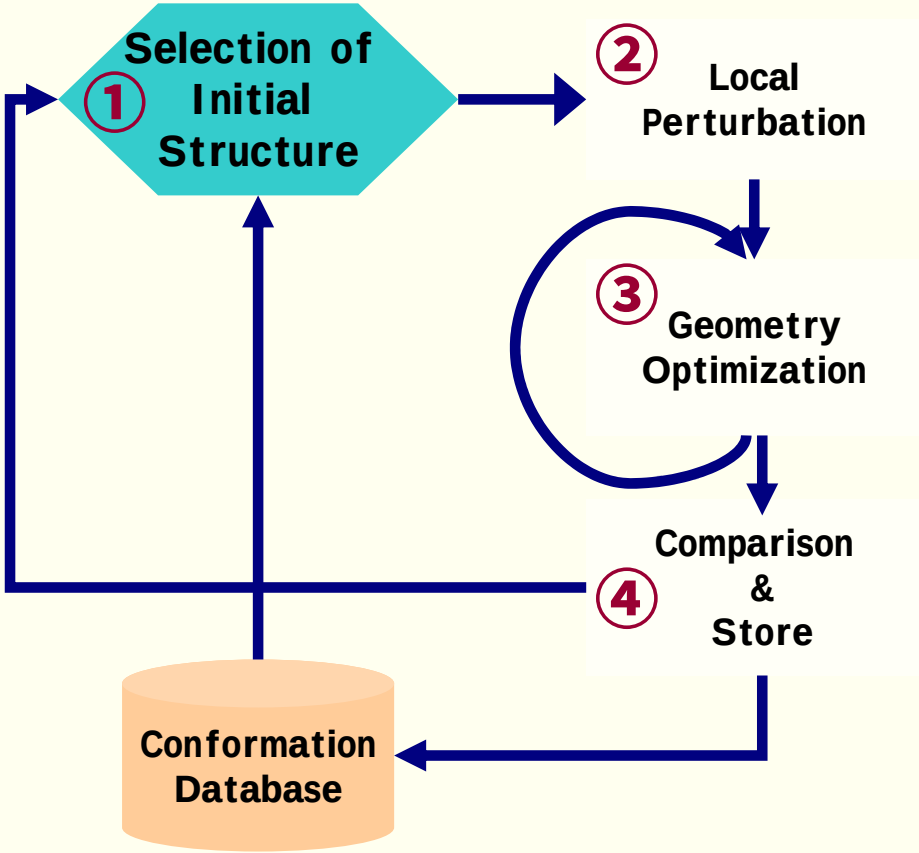


CONFLEX (2)

- **分子の配座空間を探索するプログラム**
  - **分子が取り得るすべての立体配座を自動的に発生させ，化学的に重要な配座異性体の最適化構造を探索**
  - 2D drug database → 3D structure database
- **分子力学** (Molecular Mechanics)**を使い構造最適化を行う**
  - **全原子に対応した計算が可能**
- **配座発生と構造最適化の機能を合わせ持つ**
  - **配座発生の時に使用するアルゴリズムにより優れたパフォーマンス**
  - Reservoir-Filling (**貯水池注水**) **アルゴリズム**

CONFLEXの配座探索アルゴリズム

① すでに保存されている構図の中から初期構造の選択
② 試行構造の生成
③ それぞれの構造を最適化
④ すでに得られている構造と比較し、新しい配座を保存



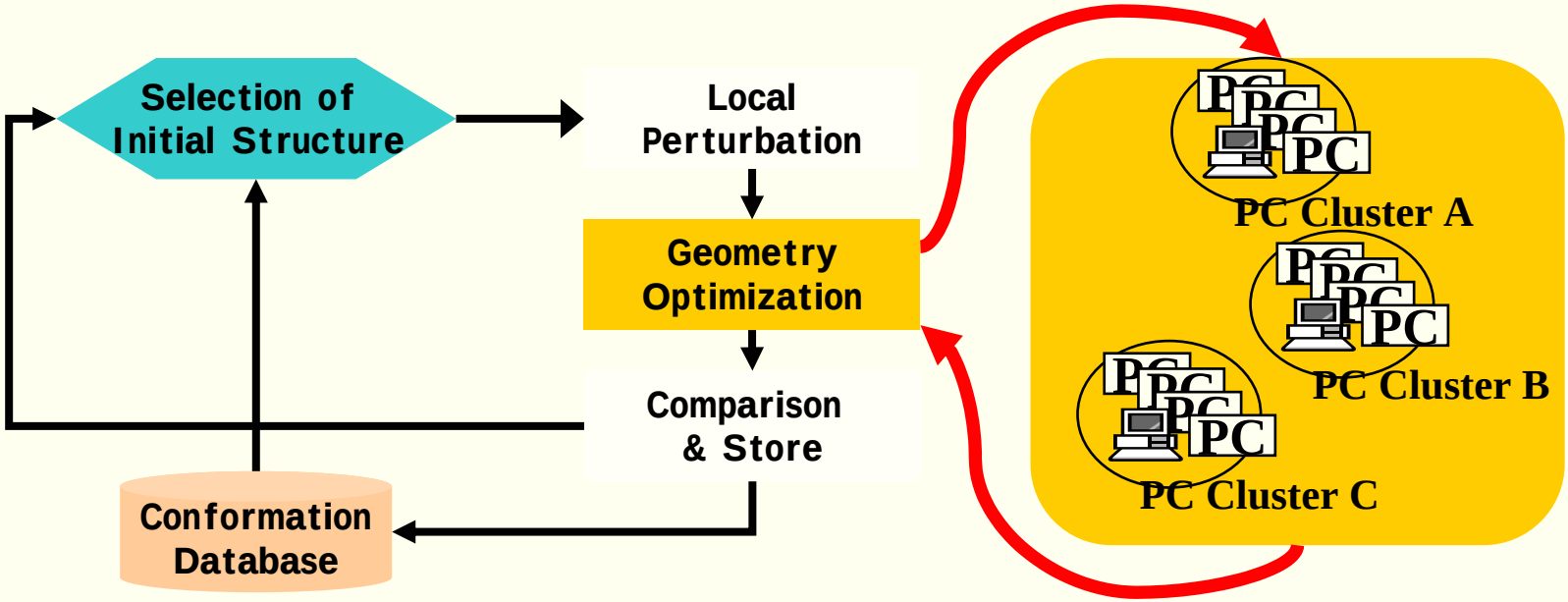
CONFLEXのグリッド化

- 大規模高分子をターゲットにするために試行構造の組み合わせの数が大きくなる
- 1つの大きな立体構造を計算によって評価するためには，多大な時間を要する
- 構造最適化の処理が全体の処理の９０％以上を占める
- これまではMPIを用いてPCクラスタで実行していた

大量の計算資源を使用できるGridで
並列計算できるように改良

CONFLEX-G:Grid enabled CONFLEX

- **全体の処理の９０%以上を占める構造最適化の処理を** Master/Worker**パラダイムで並列化**
- **巨大な計算機資源を利用することができる**
- OmniRPC**のモジュール再初期化機能を使用**
  - RPC**呼び出しごとに初期化が不要**



Our Grid Platform

Toyohashi Univ. of Tech.
Toyo Cluster
Dual Athlon 1800+
8 nodes

Tokushima Univ.
Toku Cluster
P3 1.0GHz
8 nodes

Tsukuba WAN

SINET

AIST
UME Cluster
Dual P3 1.4GHz
32 nodes

Univ. of Tsukuba
Dennis Cluster
Dual P4 Xeon 2.4GHz
10 nodes
Alice Cluster
Dual Athlon 1800+
14 nodes

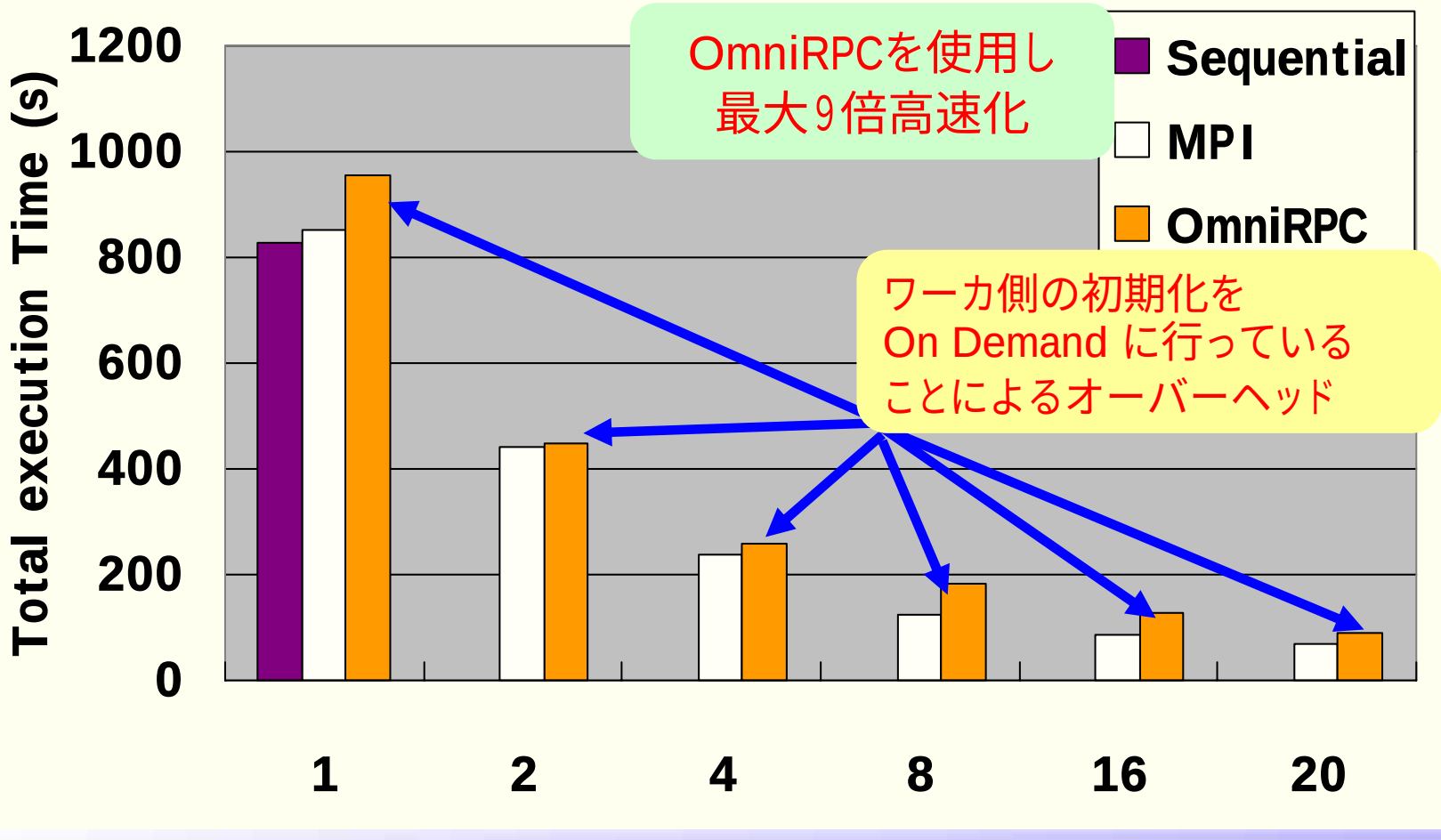
CONFLEX-Gの実験環境

- CONFLEXのバージョンは402q
- 認証にSSH, 通信の多重化はなし
- クライアントプログラムはDennisの
  マスターノードから実行
- 基本的にクラスタにあるすべてのマシンを使用して計算

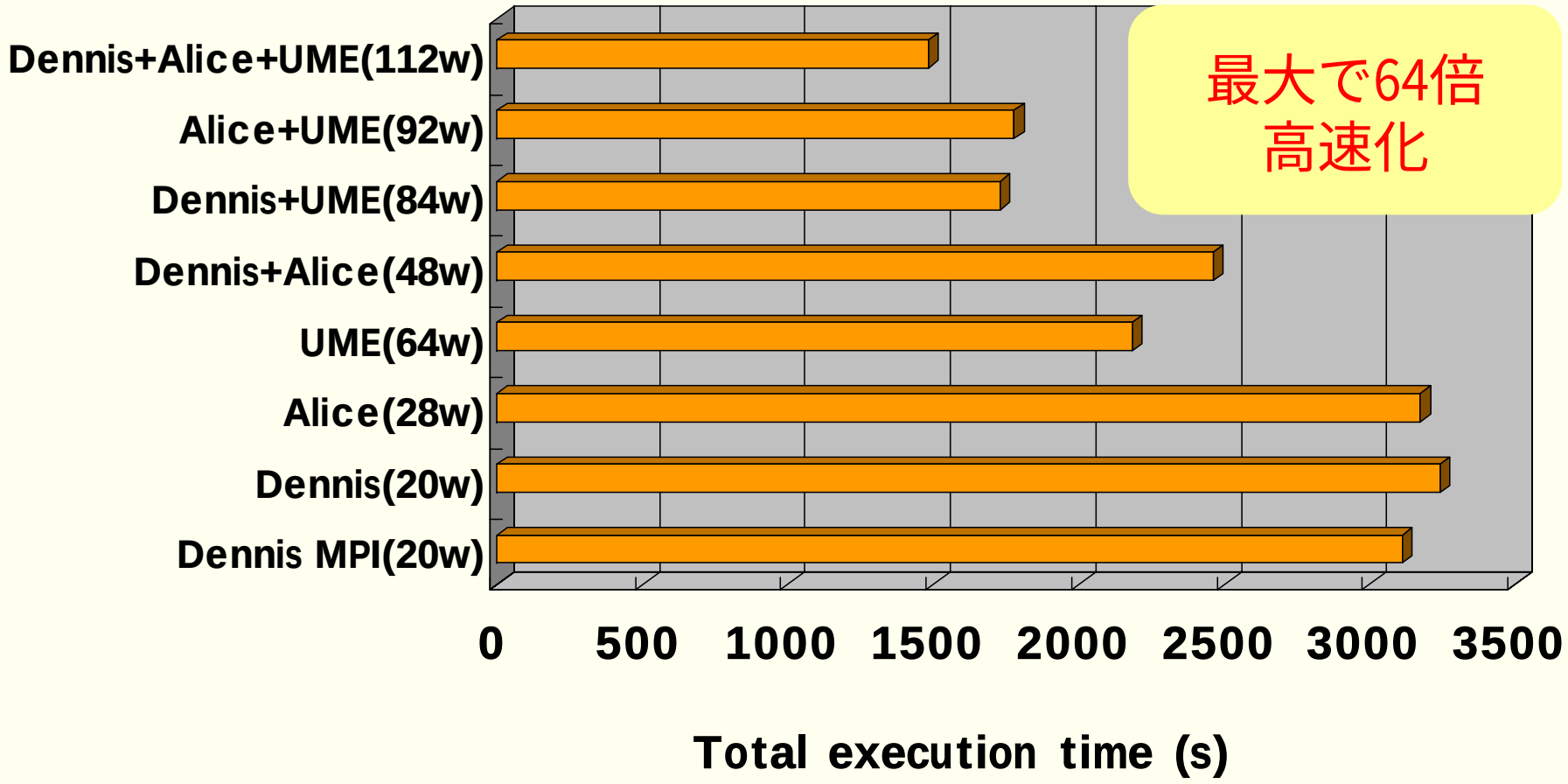
使用したサンプル分子

サンプル分子	一度に作成する試行構造数(並列度)	1試行構造最適化の平均処理時間* (s)	構造最適化を行った総試行構造数	Dennis1CPUで全試行構造最適化にかかる時間(s)
C17 (51 atoms)	48	1.6	522	835
AlaX16a (181 atoms)	160	300	320	96000 = 26.7(h)

Dennis Cluster でのOmniRPC and MPIの比較
C17 (51 atoms, 並列度48)



CONFLEX-Gの実行時間
AlaX16a(181 atoms，並列度160)



Dennis(筑波大)，Alice(筑波大)，UME(産総研)

HMCSの基本コンセプト

- **計算物理学のための計算性能は…**
  - **最先端の計算物理学においては、詳細で精密なシミュレーションのために各種物理現象を連成問題として扱う必要がある**
  - **それらの中には大規模計算のために膨大な演算量を必要とするものがある**
    - FFT: O($N \log N$)
    - **重力、分子動力学法**: O(N^2)
    - **ナノスケール物性物理**: O(N^3), O(N^4), ...
  - **通常の汎用計算機は多くの場合十分な計算性能を提供しない**
  - **何らかの形で専用計算機を導入する必要がある**

HMCSの基本概念 (続き)

- **汎用計算機**:
  柔軟なアルゴリズム とプログラミングのし易さにより様々な用途に使用
- **専用計算機**:
  絶対的な計算パワーを非常に限定された種類の処理に発揮
- 両方が必要

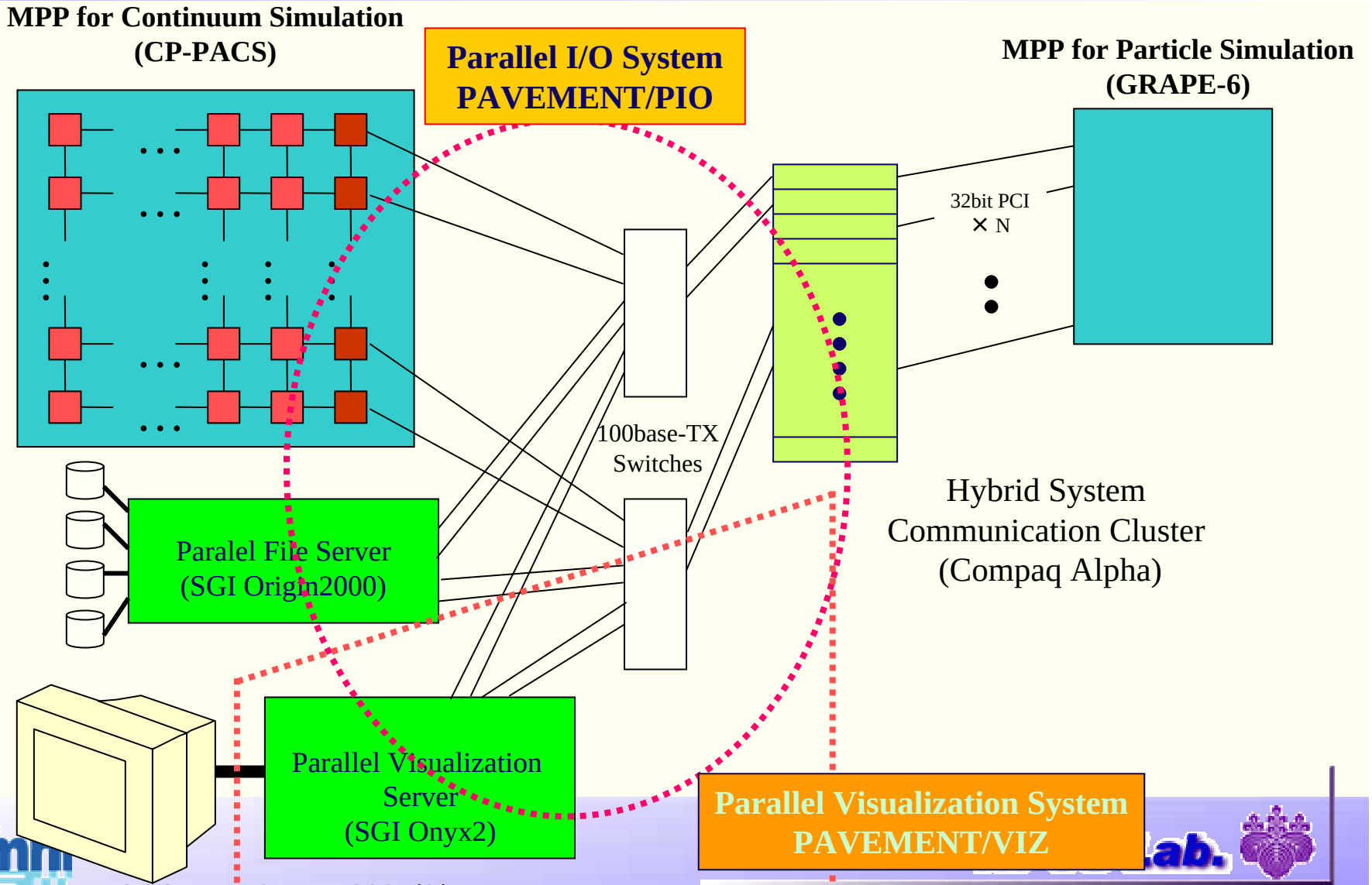
Heterogeneous Multi-Computer System
両者のタイプの計算機を
高バンド幅ネットワークで有機的に結合

Heterogeneous Multi-Computer System

- **粒子系シミュレーション (例: 重力計算) と 連続系シミュレーション (ex: 場・流体力学) を1つのシステムで統合化**
- **汎用プロセッサ (柔軟性) と専用プロセッサ (高速性)を統合**
- **汎用超並列計算機 と 専用超並列計算機 を 高バンド幅ネットワークで結合**
- **両サイドの計算機間で細かい粒度でのデータ通信 (例：毎タイムステップ) を実現**

➡ **プロトタイプシステム：CP-PACS + GRAPE-6**
(学振の「未来開拓」プロジェクトで実現)

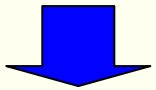
HMCSのブロックダイアグラム



HMCS-G：Grid-enabled HMCS

計算物理学におけるグリッドの意義

- **汎用の** HPC **リソース** (MPP, cluster, storage, network, etc.)**を広域ネットワークで活用**
  = **量的貢献**
- **少数のサイトに導入されている専用計算機 を広域ネットワーク上で共有化する**
  = **計算の質の向上**



HMCS-G とは Grid-RPC**を用いて広域分散された汎用計算機リソースと専用計算機リソースを結合し、**multi-physics**による次世代シミュレーションを実現する統合システム**

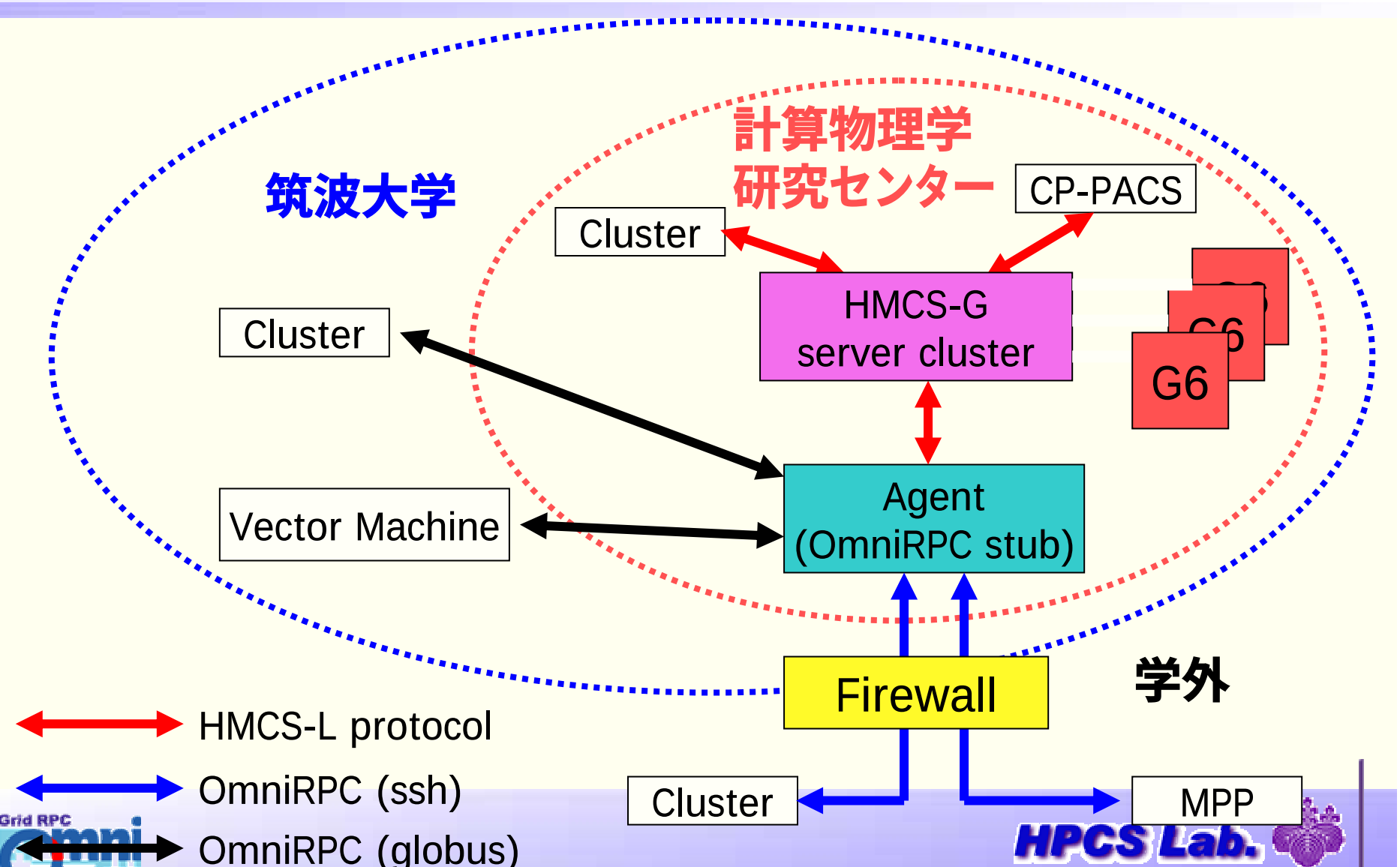
HMCS-G: Grid-enabled HMCS (for Gravity)

- **システムの目的**
  - **重力を含む計算を必要としている世界中のユーザにより専用計算機GRAPE-6を共有する**
  - GRAPE-6**を保有するサイトにおけるローカルなサービスを効率的に提供しつつリモートなアクセス要求にも柔軟に対応**
  - end-point**間の通信バッファを利用し**
    - **遅延隠蔽に適用可能な機構を提供**
    - GRAPE-6**の利用効率を高める**

HMCS-G: Grid-enabled HMCS (続き)

- **システムの実装**
  - HMCS-Lにおける汎用・専用計算機間通信プロトコルを改良し、マルチクライアントに効率的に対応するGRAPE-6サーバクラスタを構築
  - Grid用RPCシステムとしてOmniRPCを利用
    - sshまたはglobusによる認証（及びアカウンティング）
    - RPC-stubを利用した通信バッファ効果
    - リモートアクセスの処理は全てRPC-stubが吸収
      ⇒ローカルクライアントに影響を与えない

HMCS-Gのブロックダイアグラム



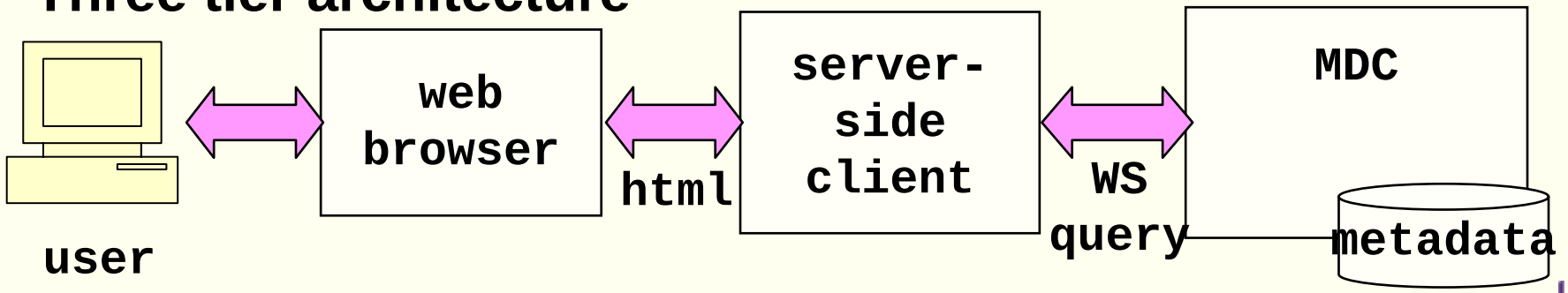
ILDG

- International Lattice Data Grid
  - **計算された**configuration**のデータの統一**
  - configuration**のデータを共有**
  - （Lattice theory**の計算のための資源の共有**）
- Working Group
  - Metadata WG: **計算された**configuration**の記述フォーマット**(QCDML)**の策定**
  - Middleware WG: WebService**を利用して、**configuration**のデータの共有のための仕組みを提供**.
    - Metadata Catalog (MDC)
    - Replica Management
    - Storage Resource Management(SRM)

Web (browser) interface

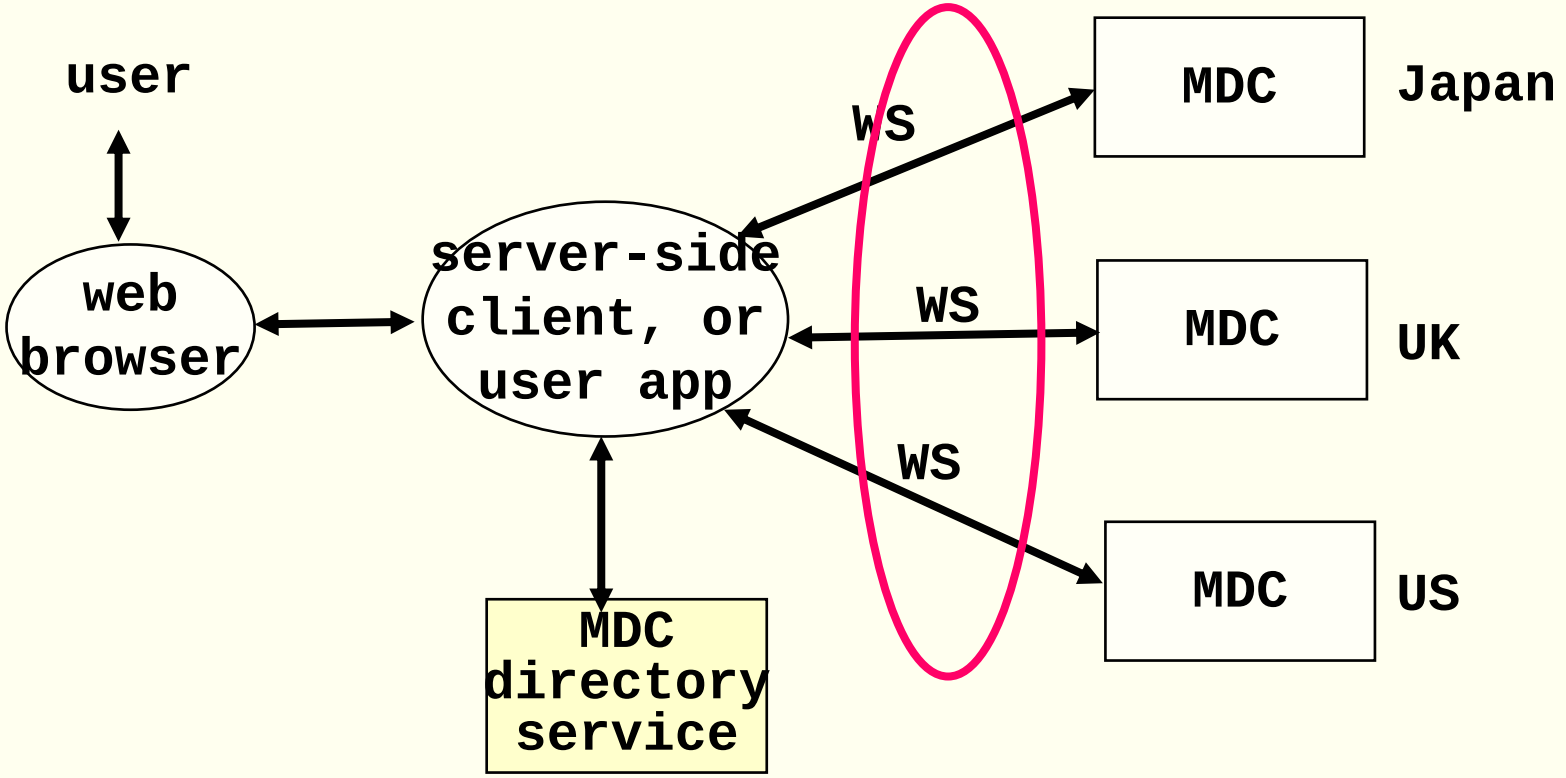
- **In many cases, users will interact with the ILDG using a standard web browser.**
  - **Some web browsers may support issuing SOAP requests, but it is planned that the ILDG collaboration will support interacting with the web services via dynamic web pages.**
  - **These pages may be implemented using server-side client such as servlets, and other techniques.**

- **Three tier architecture**



Grid-of-Grids on MDC

- Clients can access multiple MDC's at different sites
- Directory service tells the locations of MDC's, which may be managed by LDAP



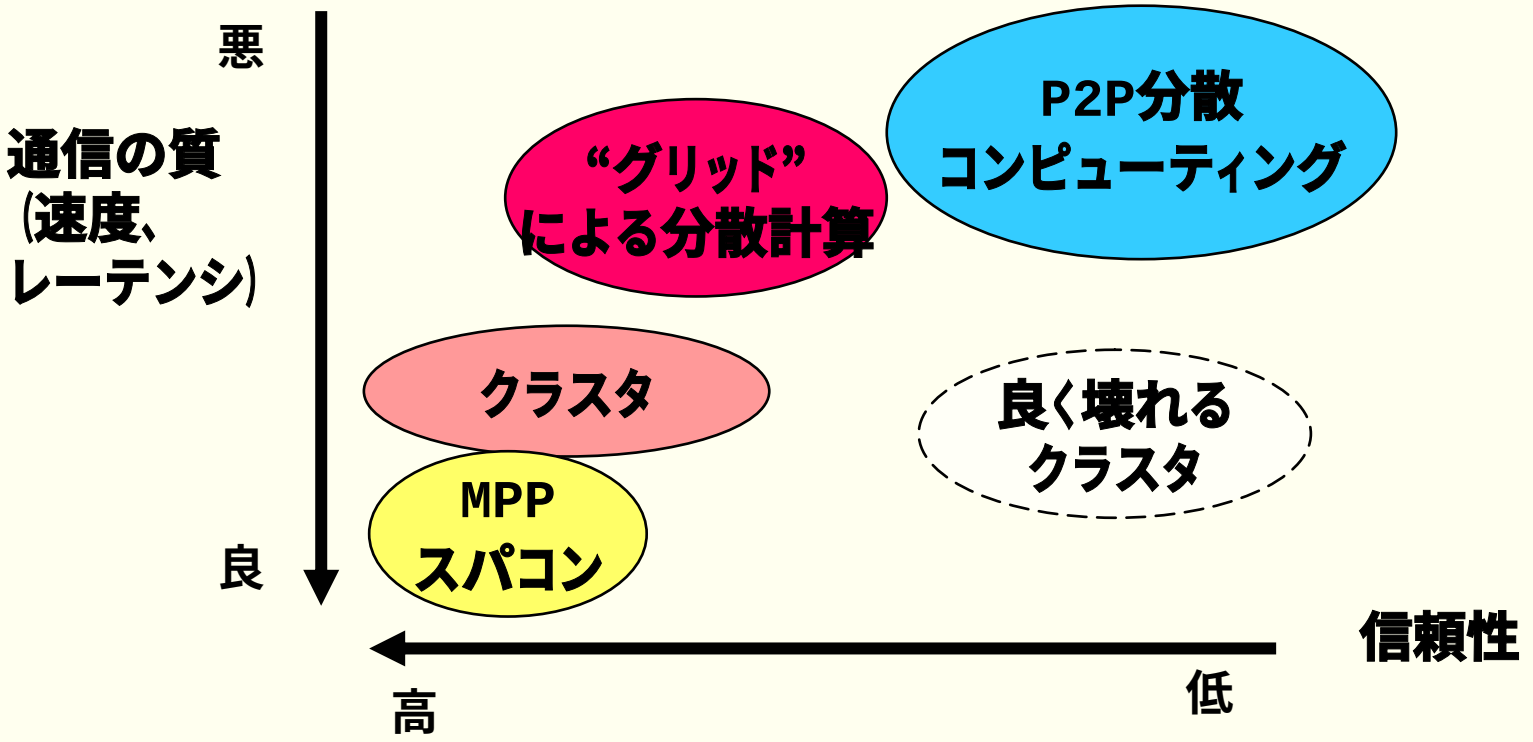
Lattice QCD Archive at CCP

- Set up and maintained by CCS
- Open to LQCD community (registration required)
- Provides the full set of CP-PACS Nf=2 configurations
  - 4 lattice spacings (a=0.2fm – 0.1fm) / 4 sea quark masses
  - 500-1000 configurations/(beta, kappa)
  - About 2TBytes total
- Currently, web browser interface only.
- Web Service interface defined by Middleware WG will be provided

http://www.lqa.rccp.tsukuba.ac.jp/

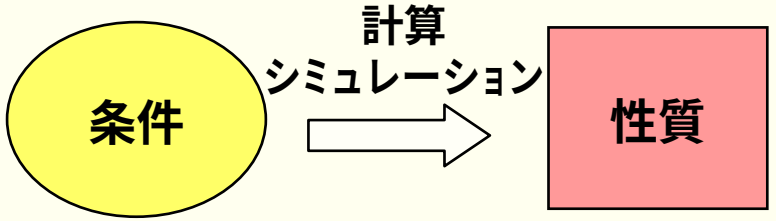
グリッドコンピューティングで何ができるか

- グリッドは、スパコンにとって変わるものではない！
  - アプリケーションをえらぶ。coarse-grain…
  - 何をさせるかは問題！…でも、パラメータサーチはたくさんある

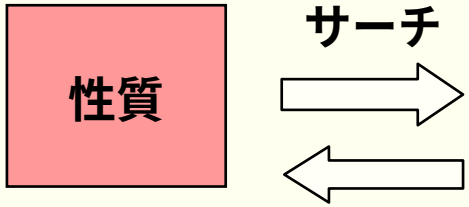


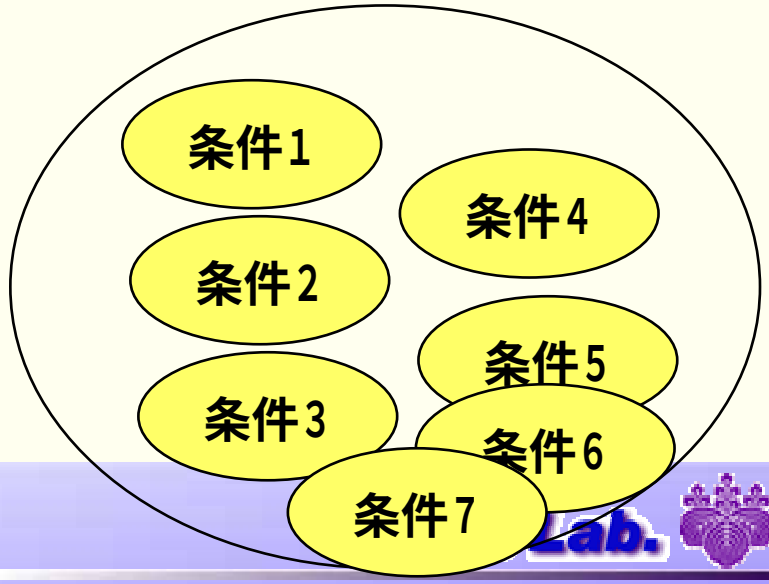
“新しい計算科学”へ、発想の転換

- **潜在的な計算資源は、たくさんある！**
  - P2Pならばオフィス・家庭にあるPCも、…1万台のPCがあれば、
    - 1GFlopsのPCがあれば、10TFlops
    - 50GBのディスクがあれば、500TB
- **従来の計算科学**
  - ある条件で、どのような現象が現れるか。
- **膨大な計算リソースが使えれば、…**
  - ある性質を持つ条件は何か。
  - 設計



パラメータサーチ
アンサンブル計算





“新しい計算科学”へ、発想の転換

- **グリッドコンピューティングの計算の性質に適した新しいアルゴリズム、発想が必要！**

- **例**：Folding@home
  - **スタンフォード大学のビジャ・パンデ助教授（http://www.stanford.edu/dept/chemistry/faculty/pande/）らが開発した“確率分割法”**
  - **一般的なMD計算では、たん白質を周期境界条件で区切って“空間分割”によって並列化。この手法ではCPU間通信がボトルネックとなり、並列度をあまり上げられない**
  - **“確率分割”では、各CPUに薬物分子とたん白質とのクラスター構造を与え、周囲の水分子の状態などの計算条件を変えたシミュレーションを並列で実行させる。（パラメータサーチ！）**

- folding**問題も、グリッド（**P2P)**で可能になった！？**

“新しい計算科学”へ、発想の転換

- データグリッド
  - 様々な電子的なデータが蓄積しつつある
    - 加速器
    - 衛星データ
    - 天体観測
    - センサーデータ
    - ゲノム
  - 例：Gfarmによる天文データの解析
    - Gfarm： 産業総合研究所で開発されたデータグリッドのためのミドルウエア
    - 膨大な天体観測の画像データを処理し、解析
  - データマイニング技術による発見
  - 分散されたデータの統合

- データグリッドの成功のためには、分野のコミュニティのデータの共有利用のための国際的な協力が必要
  - ILDG

- **OmniRPCのweb page**

http://www.omni.hpcc.jp/omnirpc/