

Ninf-Gを用いたグリッド上の 最適化問題計算

合田 憲人(東京工業大学)

藤沢 克樹(東京電機大)

小野 功(東京工業大学)

松岡 聡(東京工業大学)



最適化問題と グリッドコンピューティング

最適化問題

与えられた制約条件 S のもとで目的関数 $f(x)$ を最小(最大)にする解を求める.

minimize/maximize $f(x_1, x_2, \dots, x_n)$

subject to $(x_1, x_2, \dots, x_n) \in S$

S : 実行可能解

様々な工学分野(OR, 制御, 情報, ...)で用いられる.

グリッドコンピューティングによる 最適化問題計算

■最適化問題計算

- 大規模問題の計算には莫大な計算パワーが必要

 - 計算が諦められる場合も多い.

- 多くの問題に見られる特徴

 - 全体の計算は依存関係の無い(少ない)大量のタスクから構成される.

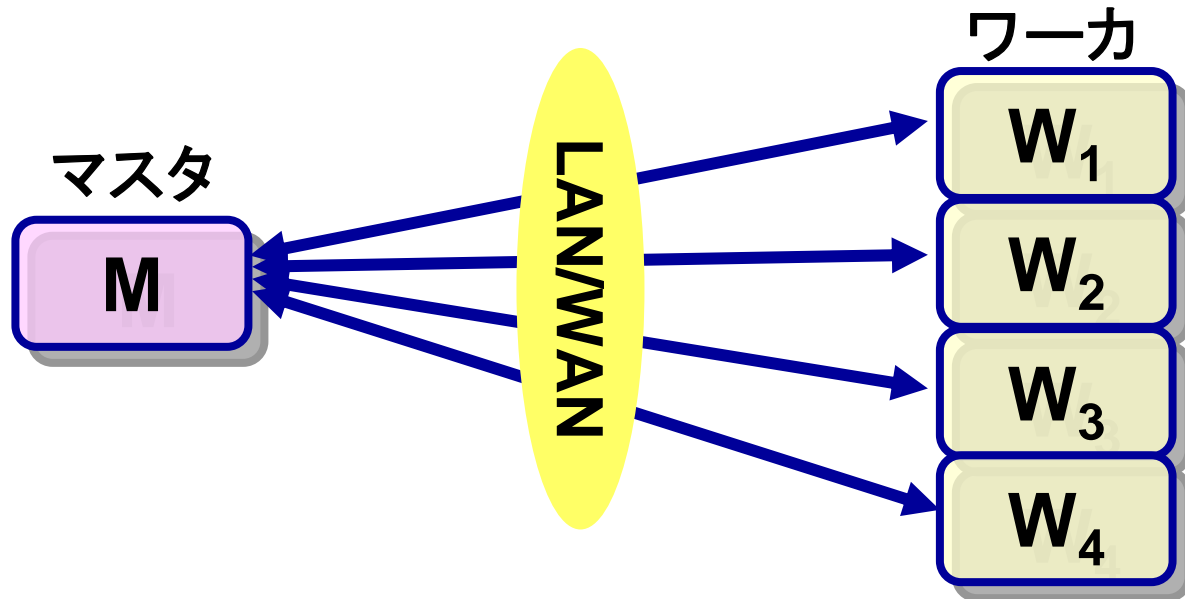
■グリッドコンピューティングへの期待

- 異なるタスクを並列に実行可能.

- グリッド上の大量の計算機を用いることにより大幅な計算時間短縮が期待される.

マスタ・ワーカ方式

- 1台のマスタプロセス(マスタ計算機)が複数のワーカプロセス(ワーカ計算機)を統括.
 - タスクのアイドルワーカへの割り当て
 - 計算結果回収
- 並列計算でよく使われる並列実行方式



Ninf-Gによる
実装が容易

グリッド上でのマスタ・ワーカ方式の問題

■ 通信オーバーヘッドの増大

- グリッドを構成するネットワークはWANである.

 - WAN上の遅延, スループット低下

- グリッド上の計算資源の多くはファイアウォールにより守られている.

 - ファイアウォール通過処理に伴うオーバーヘッド

■ 単一マスタによる性能ボトルネック

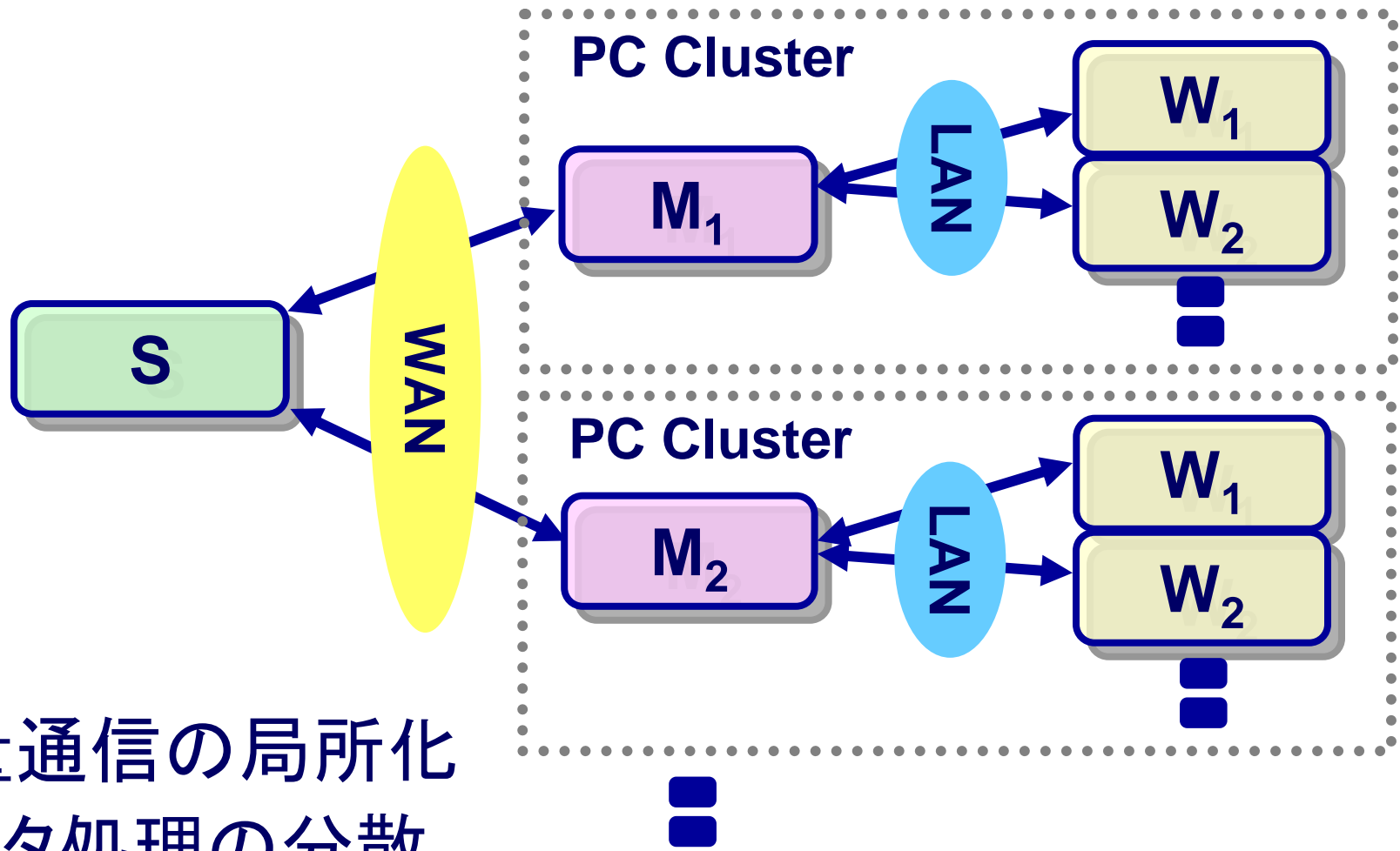
- 1台のマスタが統括できるワーカ数には限界がある.

階層的マスタ・ワーカ方式

スーパーバイザ

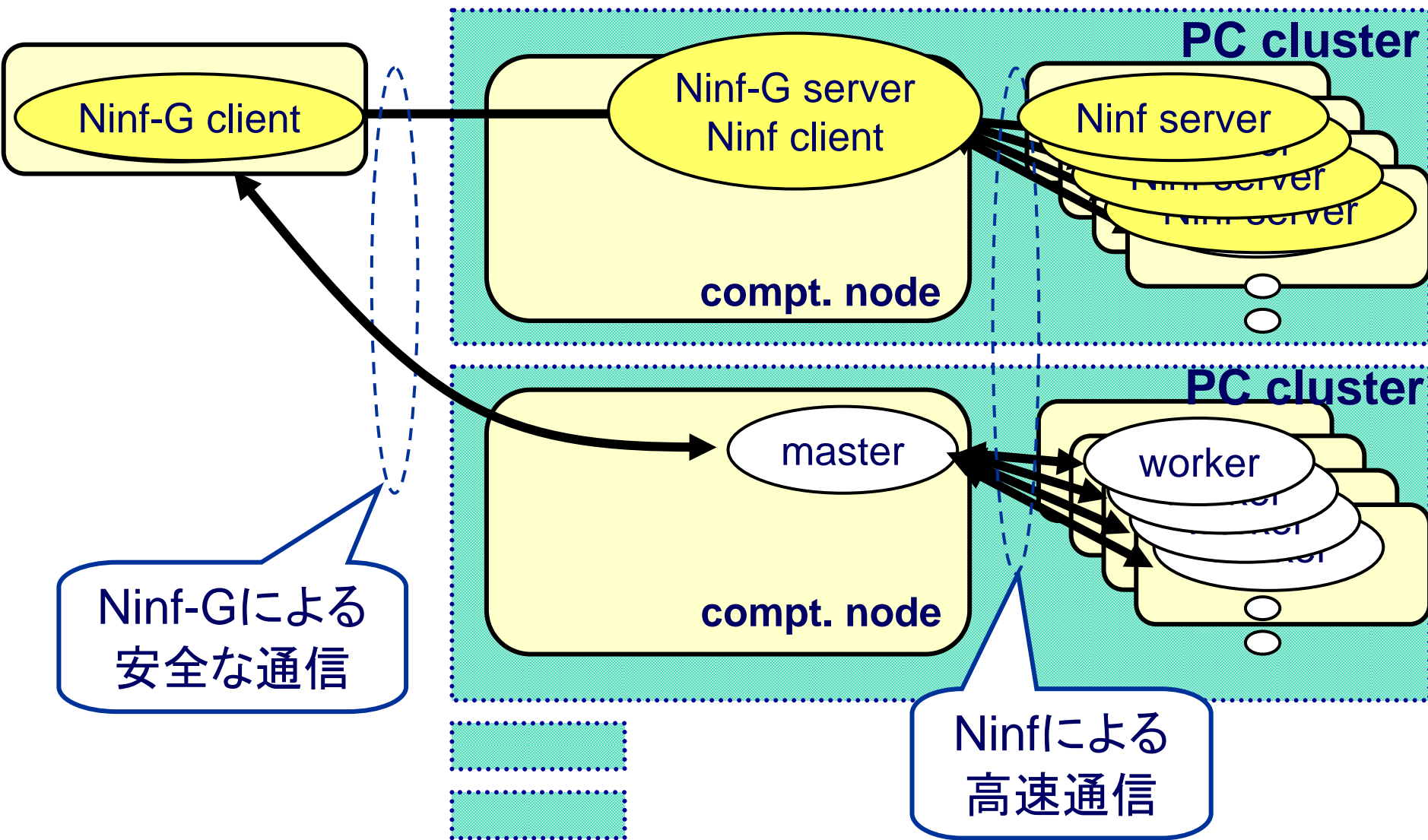
マスタ

ワーカ



- 大量通信の局所化
- マスタ処理の分散

GridRPCによる実装

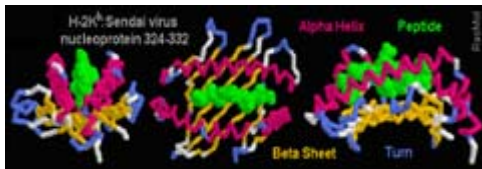
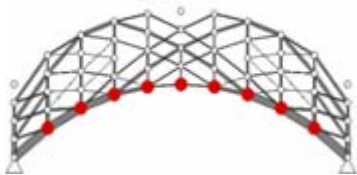


事例

1. 遺伝アルゴリズムによるNMR蛋白質立体構造決定（小野@東工大）
2. サプライ・チェーン最適化システム（藤澤@東京電機大）
3. 並列分枝限定法（合田@東工大）

ACT-JST Testbed

- 大規模最適化問題計算を主目的としたグリッド実験環境
 - 1000CPU規模の実験環境構築
 - アプリケーション開発
 - グリッド研究者と最適化問題研究者の協調



TDU



AIST



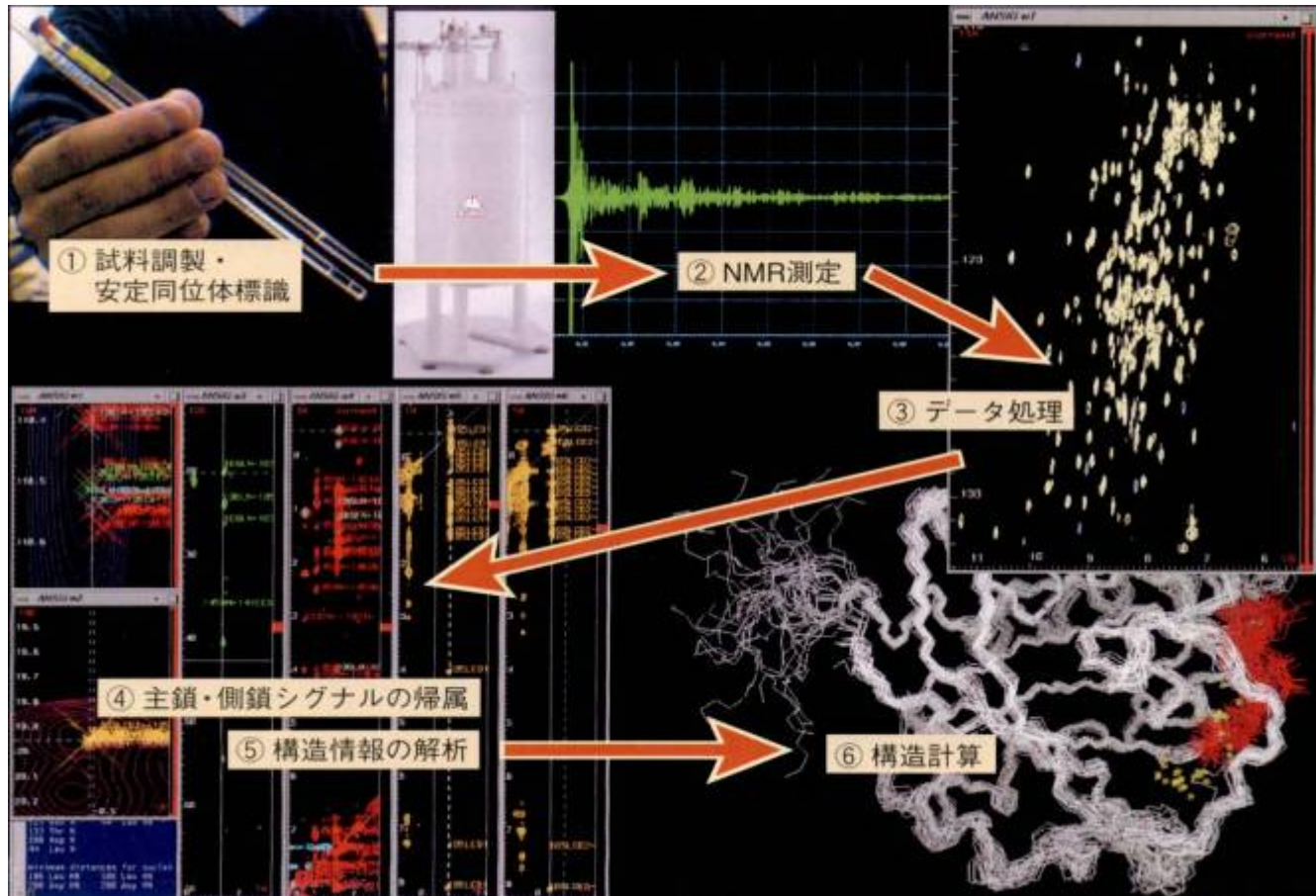
TITECH



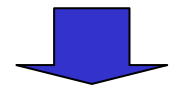
Tokushima U.



事例1: 遺伝アルゴリズムによるNMR蛋白質立体構造決定



専門家でも、「構造情報の解析」に数ヶ月以上の試行錯誤を要する！

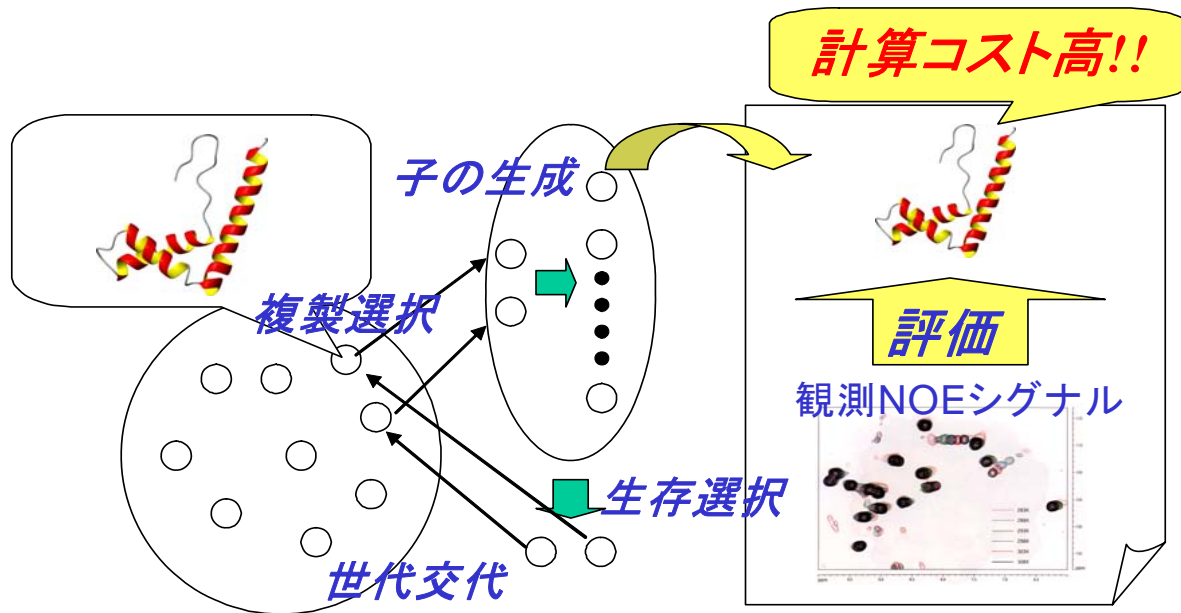


自動化
高速化
が望まれる

[source: Isao Ono, TITECH]

NMR蛋白質立体構造決定のための遺伝アルゴリズム [Ono 02]

遺伝アルゴリズム(GA)に基づく解法



13残基の α -helix
構造決定問題において、
専門家に迫る構造を**自動的**
に求めることに成功!!

大規模立体構造決定での問題点

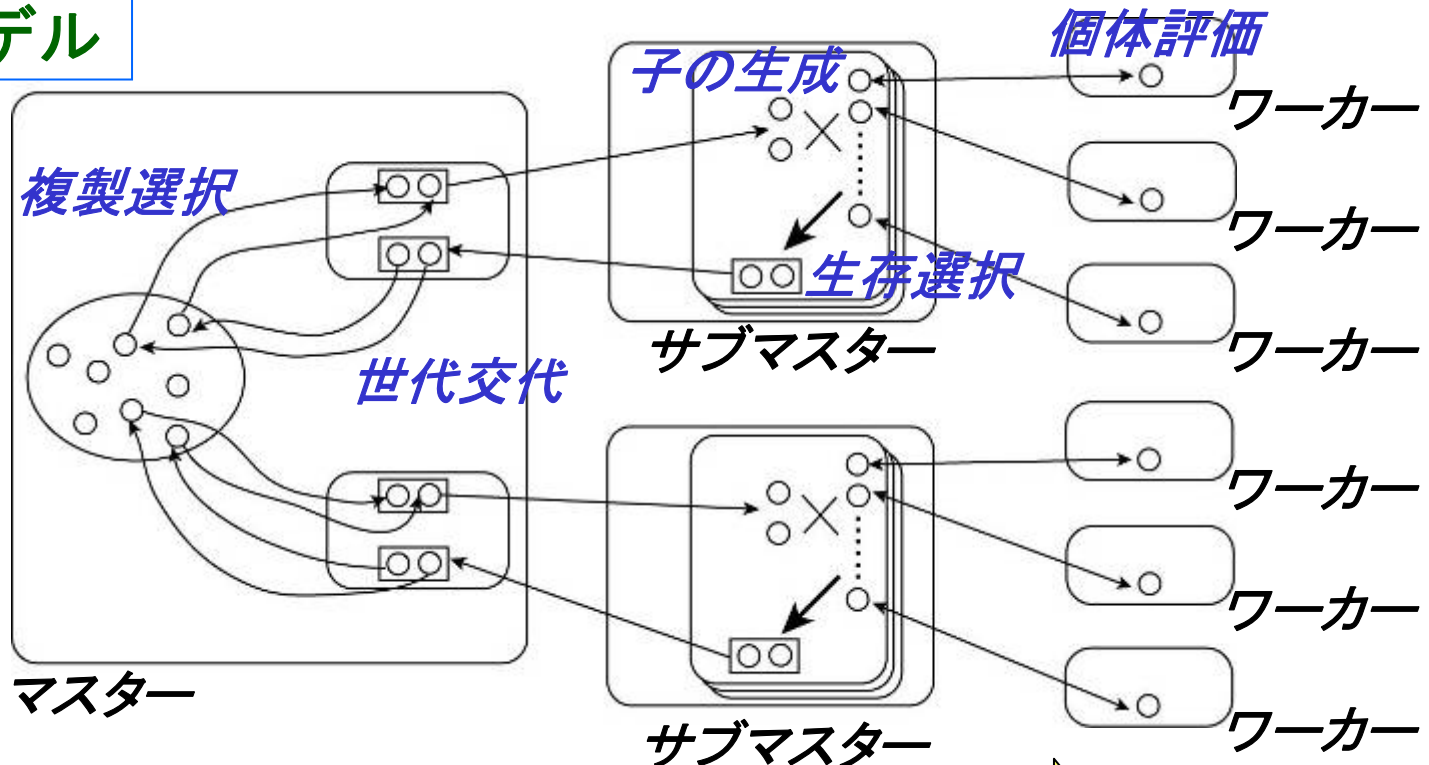
78 残基立体構造決定では、Pentium III 1.4GHz/1CPUで**約200日**かかる

現実的な時間での計算完了には、**数百~1000CPU**程度での並列計算が必要

→ **高速化のためグリッド向けに並列化**

NMR蛋白質立体構造決定のためのグリッド 向けGAシステムの提案

提案モデル



個体評価の計算時間コストが高く、独立に計算可能 → 評価計算を並列化

1,000CPU程度のスケーラビリティに対応

・ 世代交代ループを複数同時に実行

→ 並列度向上

・ マスターの機能をマスターと複数のサブマスターに分散

→ 負荷分散

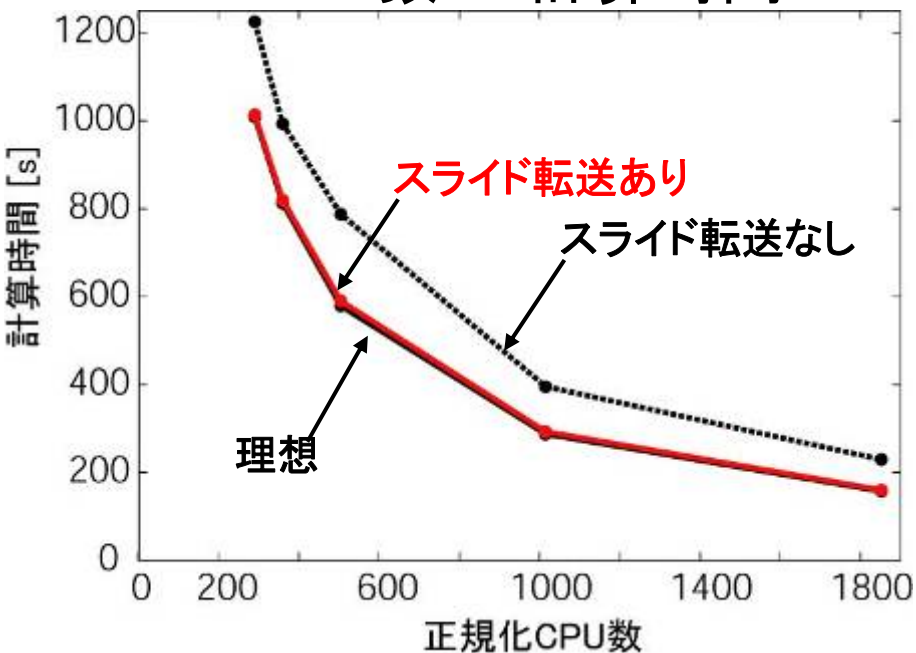
実験

試行回数: 独立3試行の平均 打ち切り世代数: 500 生成子個体数/1世代: 200

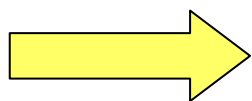
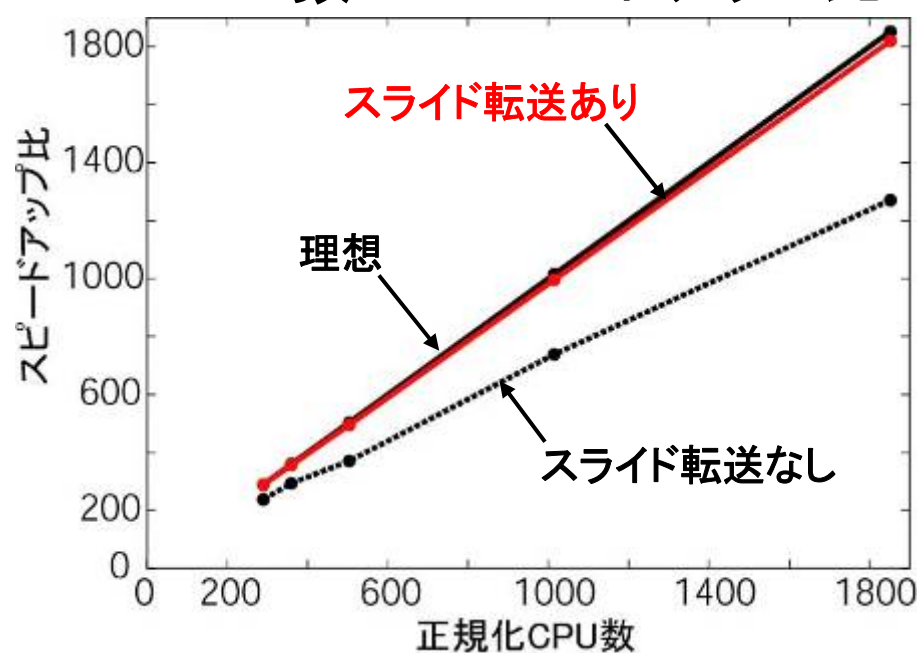
集団サイズ: 500 世代交代スレッド数: 10, 14, 16, 36, 50/サイト数で増加

正規化CPU数: Pentium III 1.4GHz/1CPUの平均評価性能を1.0として正規化

CPU数 vs 計算時間



CPU数 vs スピードアップ比

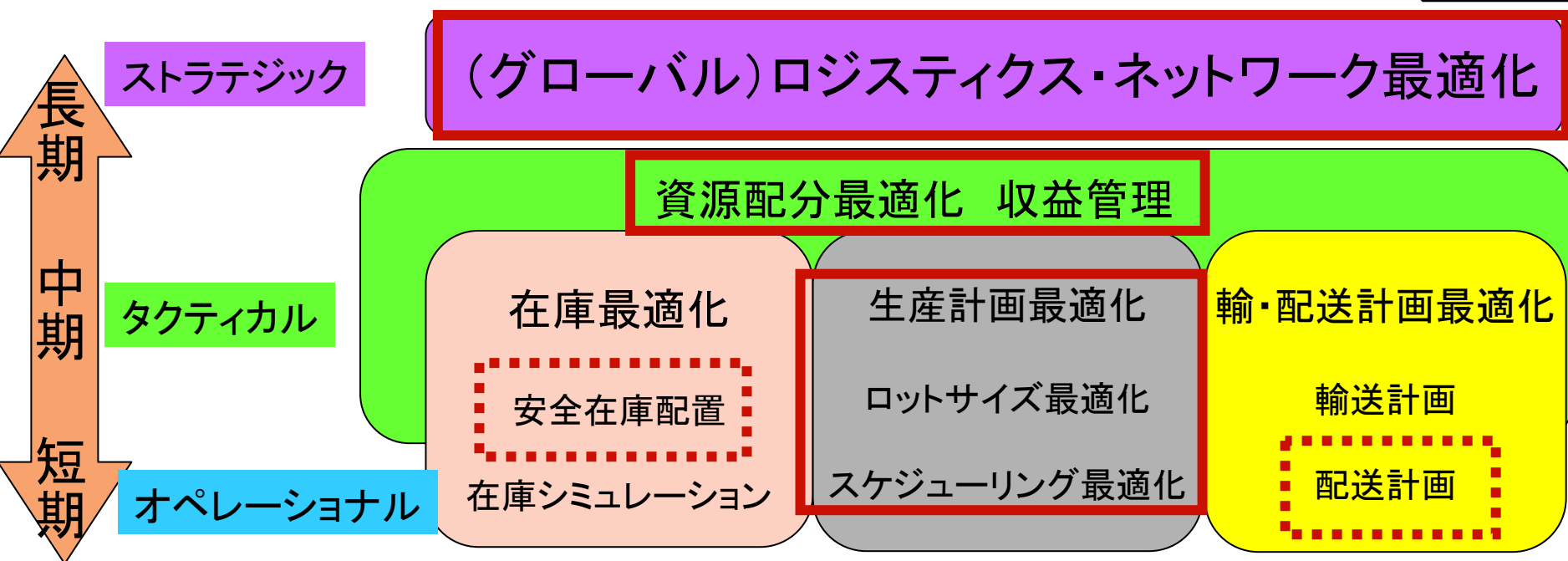
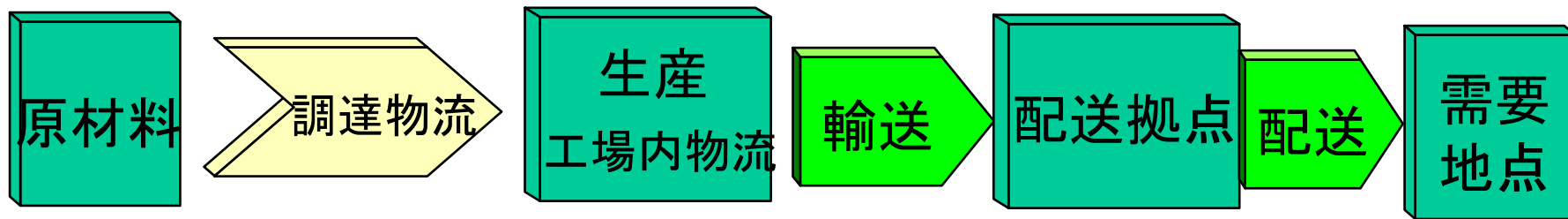


**1,196CPUで提案システムが
理想に近いリニアなスケラビリティ!!**

Kento Aida, Tokyo Institute of Technology

[source: Isao Ono, TITECH]

事例2: サプライ・チェーン最適化



ロットサイズ最適化: 1品目の例題

期(日, 週, 月, 時間): 1, 2, 3, 4, 5(5日分)



● 段取り □ 生産

段取り費用: 3万円

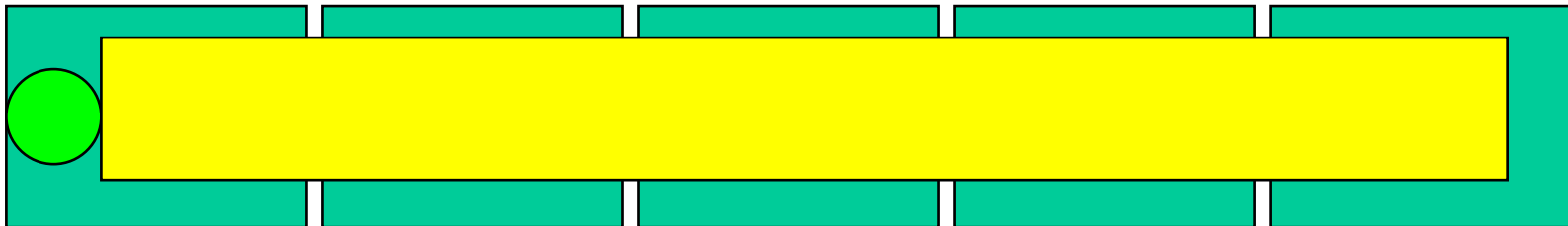
需要量 : 5, 7, 3, 6, 4 (日あたりの必要量; トン)

在庫費用 : 1日あたり1万円

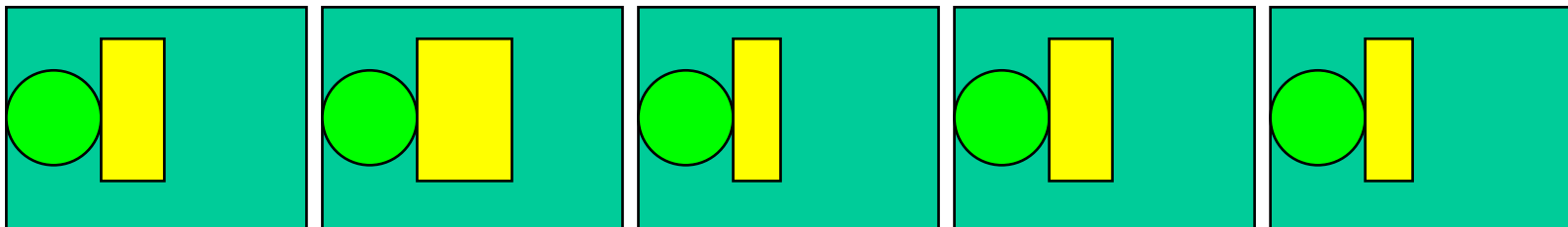
生産費用 : 1, 1, 3, 3, 3 (1トンあたり)

ロットサイズ最適化：1品目の例題(続)

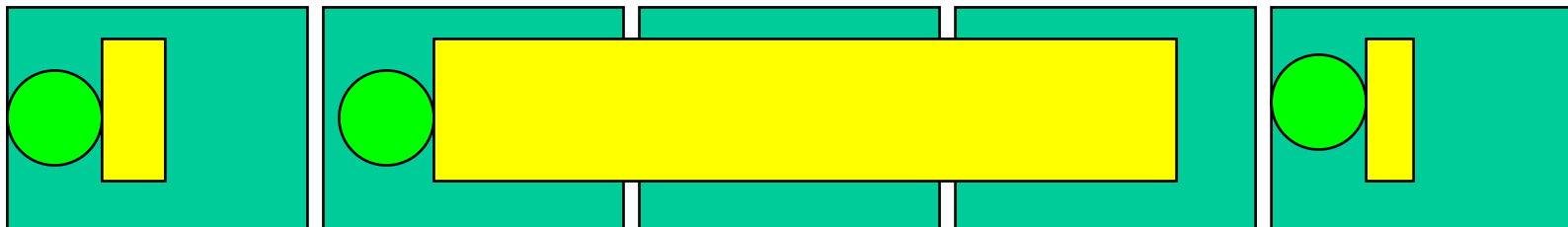
一度に生産：段取り(3) + 生産(25) + 在庫(20 + 13 + 10 + 4) = 75



JIT生産：段取り(15) + 生産(51) + 在庫(0) = 66



最適生産：段取り(9) + 生産(33) + 在庫(15) = 57



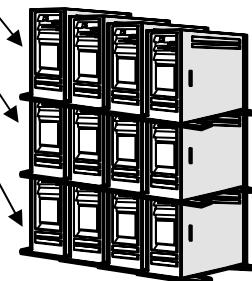
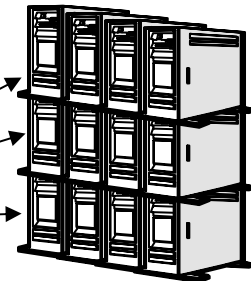
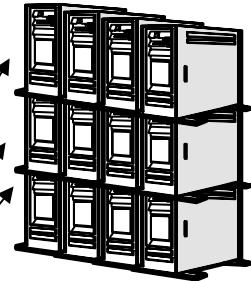
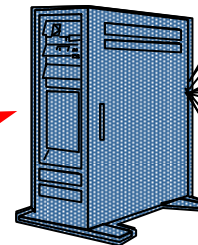
1. 在庫方策最適化: WebInv
2. 安全在庫配置: WebSCM
3. ロジスティックネットワーク最適化: WebDesign
4. ロットサイズ最適化: WebLot
5. スケジューリング最適化: WebSeq
6. 配送計画最適化: WebMETRO
7. 需要予測システム: WebForecast
8. 収益管理最適化: WebRM

Ninf1-client

&

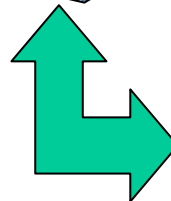
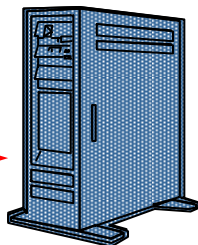
NinfG-server

Ninf1-server



MPI
並列化

Webクライアント



Windows 2003 Server

- IIS+ASP.NET
- MySQL
- Cog+Apache Ant
- Java SDK
- NinfG-client

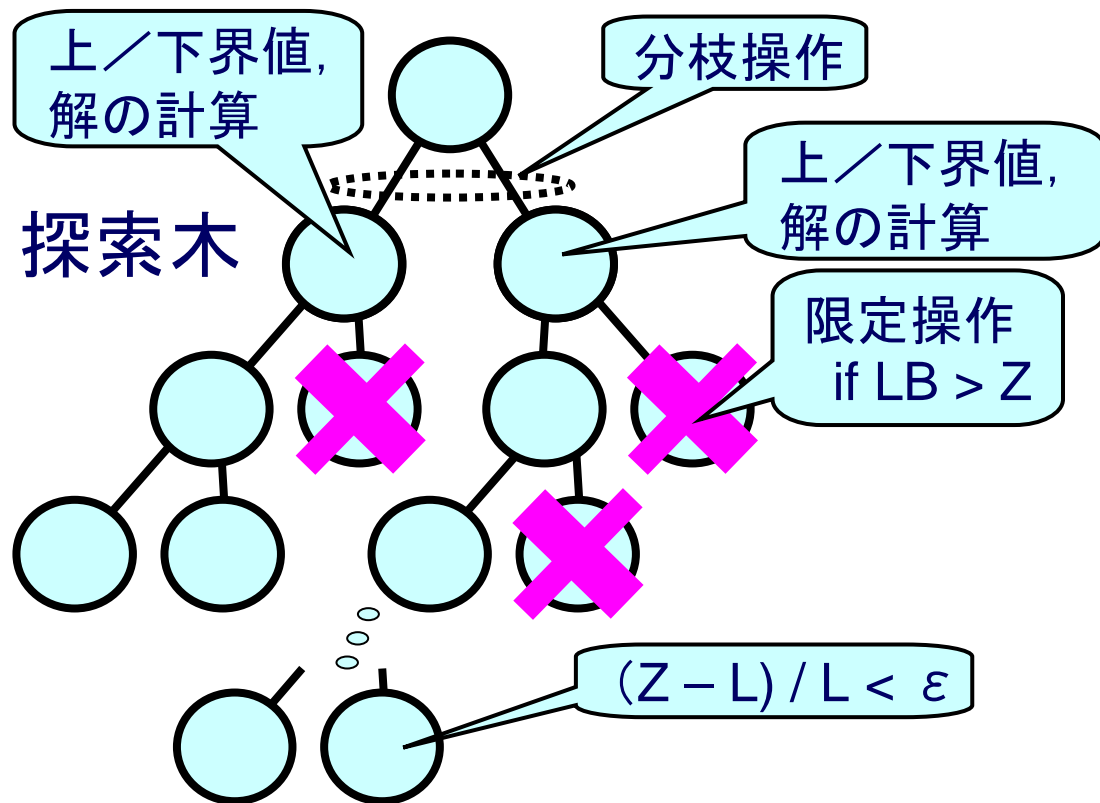
事例3:

分枝限定法

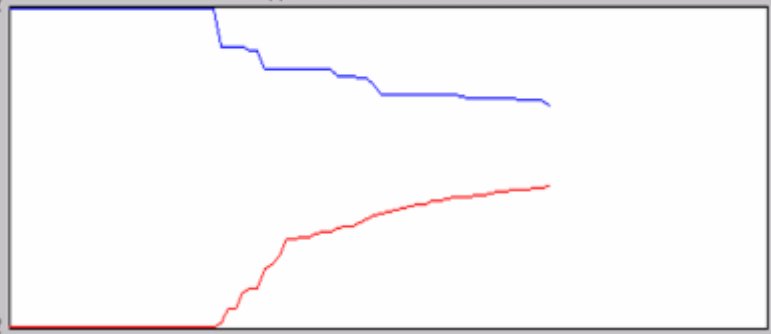
■最適解の探索

- 問題を複数の小規模な問題(子問題)に再帰的に分割して, 各子問題について解の計算を行う.
- 最適解の存在しない子問題(例: 下界 > 暫定値)は探索木から削除.

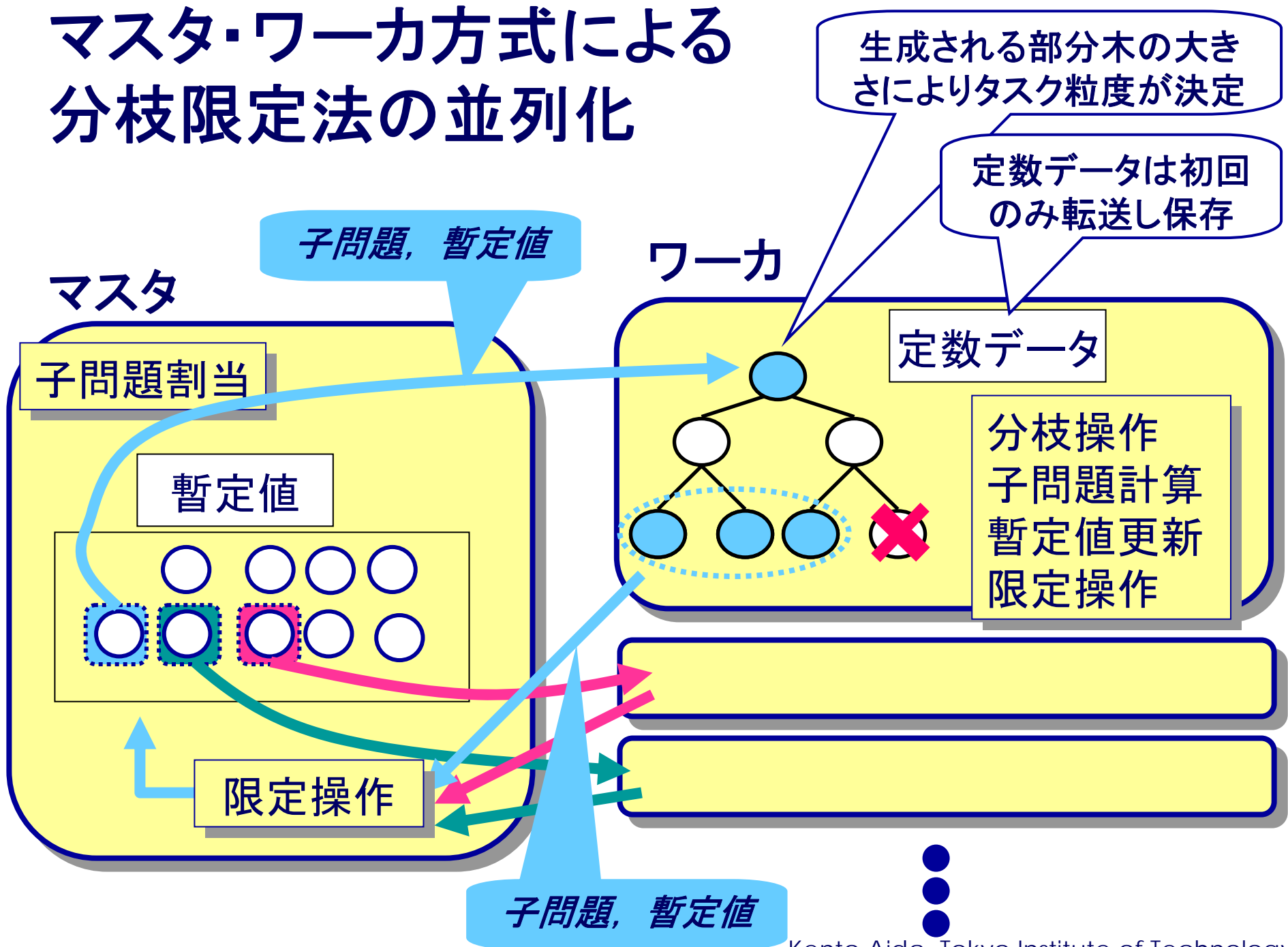
- 暫定値 = 子問題間の最小上界値 (最小化問題の場合)



best upper bound & minimum lower bound

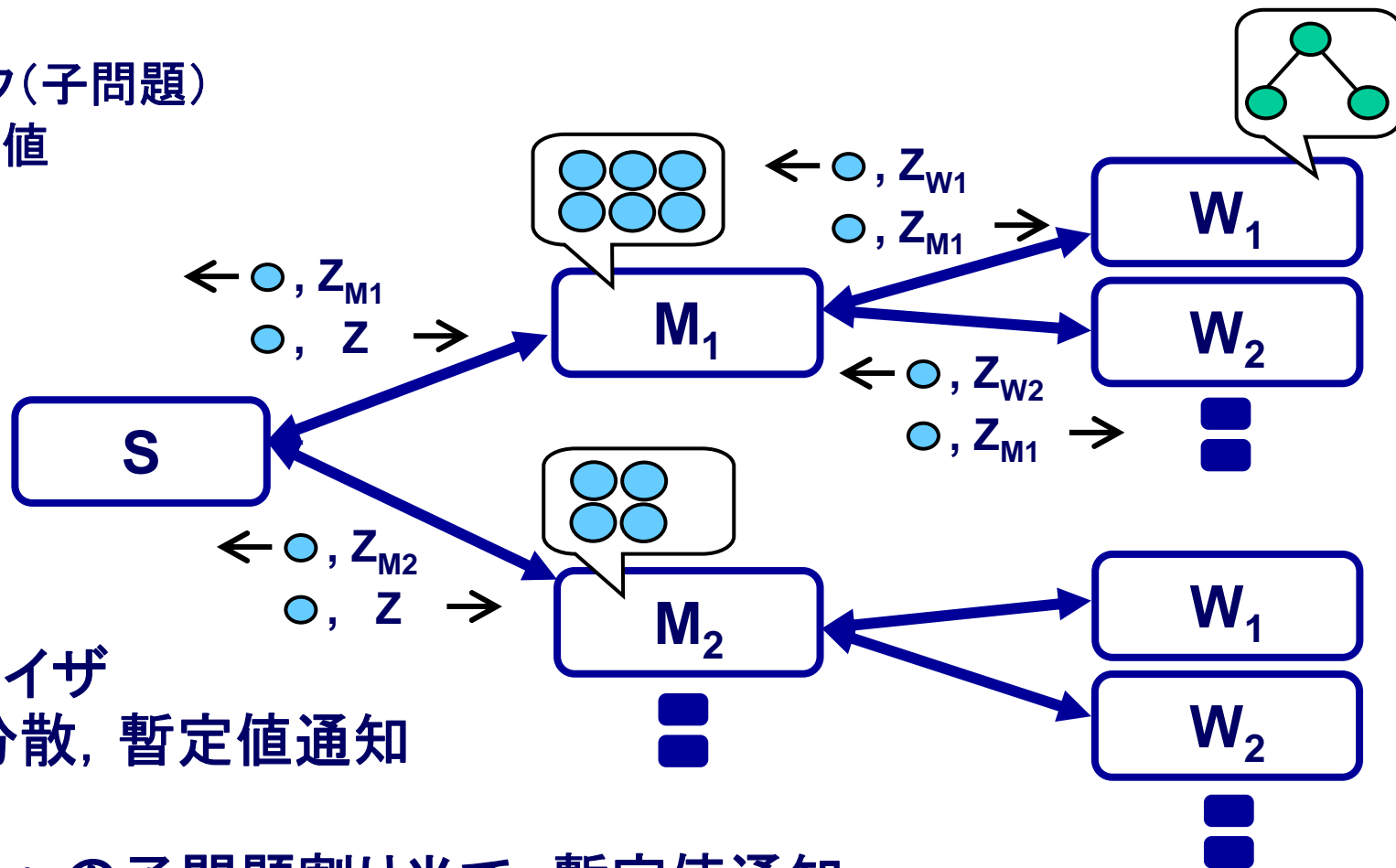


マスタ・ワーカ方式による 分枝限定法の並列化



階層的マスタ・ワーカ方式を用いた 並列分枝限定法

● : タスク(子問題)
Z : 暫定値



□ スーパーバイザ

負荷分散, 暫定値通知

□ マスタ

ワーカへの子問題割り当て, 暫定値通知

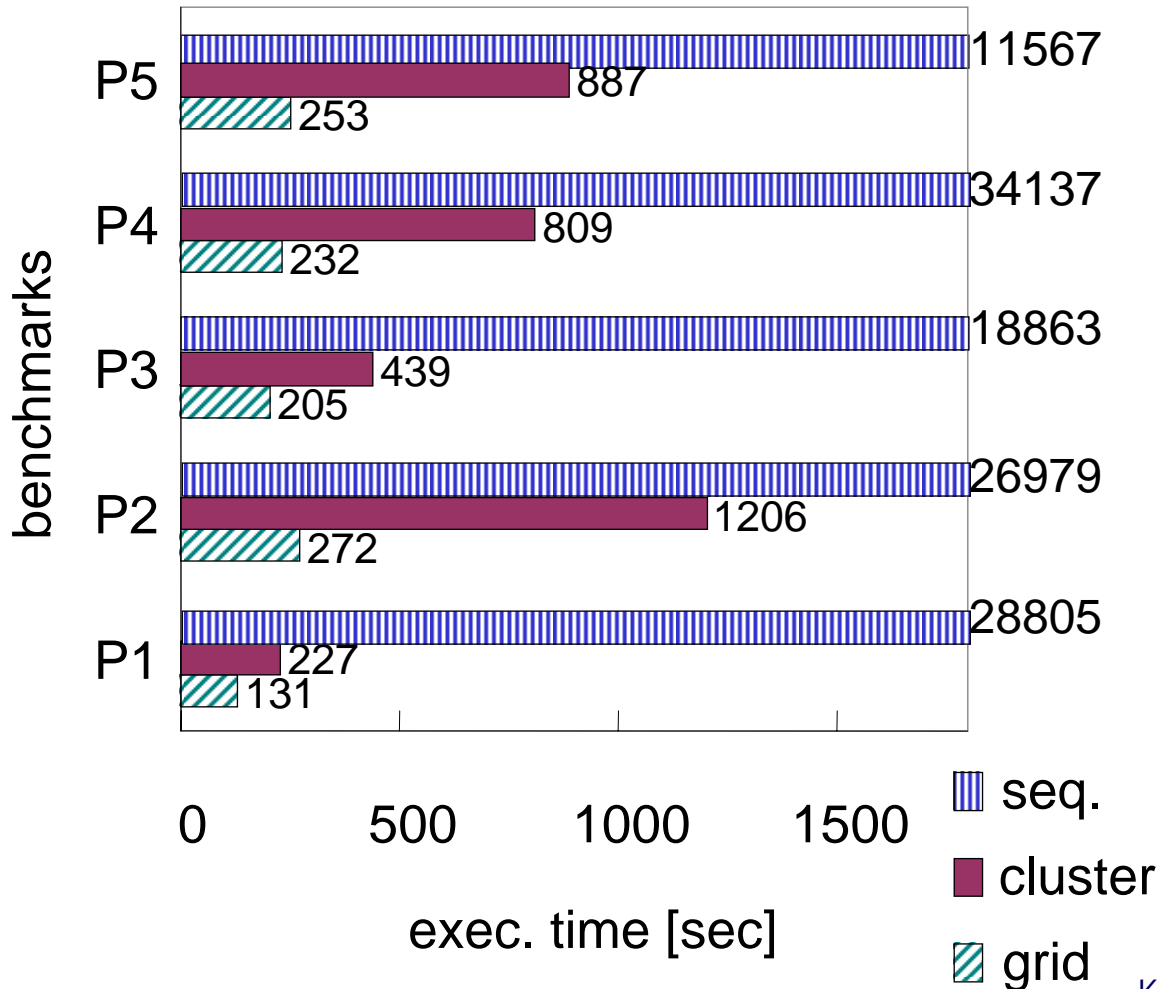
□ ワーカ

子問題計算

グリッド上でのアプリケーション実行時間

BMI固有値問題

P1-P5: $n_x=6, n_y=6, m=24$,
タスク実行時間 = 0.5sec程度



seq: Blade (1CPU)
cluster: Blade (73CPUs)
grid: 4sites (412CPUs)

- 例: P2
逐次計算: 9.5h
グリッド計算: 4.5min
- 細粒度アプリに対する階層的マスター・ワーカ方式の有効性を確認.

ポータル

遺伝アルゴリズムによる NMR蛋白質立体構造決定

The screenshot displays a web browser window with two tabs. The active tab is titled "The output page of best individual - Microsoft Internet Explorer" and shows a URL: "https://ortoros.is.tokushima-u.ac.jp/8443/TarpakuWeb1101". The page content includes the heading "Global Optimization of Protein 3-Dimensional Structures in NMR". Below this, there is a section titled "globus.is" with a list of server status information:

Name	Status	Call
pad014	DOWN	
pad024	DOWN	

On the right side of the browser window, there is a list of links under the heading "PCK Data":

- [id_ite_0_pck.zip](#)
- [id_ite_1000_pck](#)
- [id_ite_2000_pck](#)
- [id_ite_3000_pck](#)
- [id_ite_4000_pck](#)
- [id_ite_5000_pck](#)
- [id_ite_6000_pck](#)
- [id_ite_7000_pck](#)
- [id_ite_8000_pck](#)

The central part of the browser window shows a 3D visualization of a protein structure, rendered as a red and yellow ribbon. The structure is complex, with several alpha-helices and beta-strands. The browser interface includes standard navigation buttons (back, forward, search, etc.) and a status bar at the bottom indicating "ページが表示されました" (Page displayed) and "インターネット" (Internet).

[source: Isao Ono, TITECH]

Kento Aida, Tokyo Institute of Technology

ロットサイズ決定問題

[WebLotメイン] [品目] [資源] [品目・期] [品目・資源] [品目対] [資源・期] [問題選択メニュー]

ロットサイズ決定システム WebLot メイン

最適化 最適費用 1460

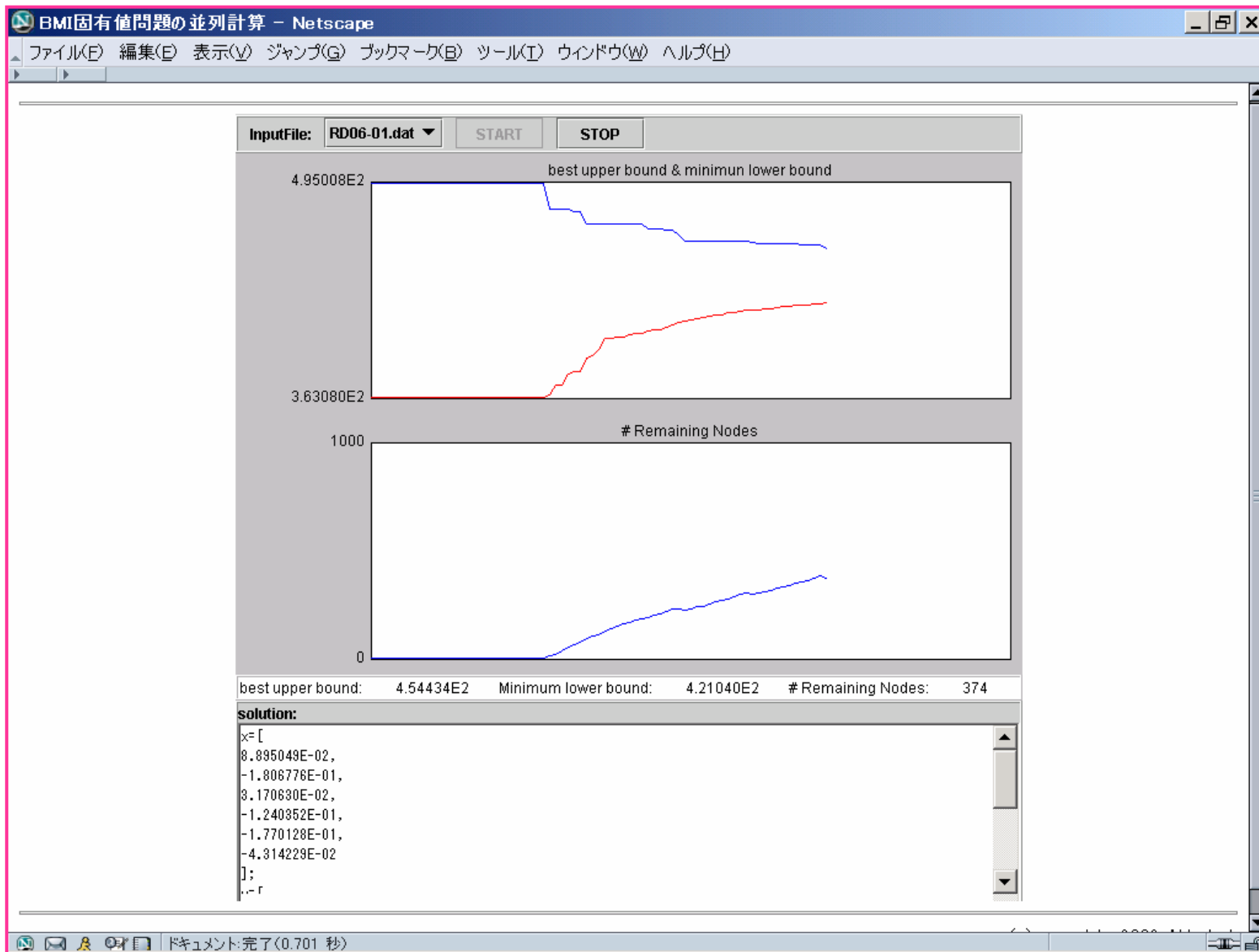
Methods CapacityScaling RelaxAndFix BranchAndBound

[需要量:段取り費用:段取り時間:変動費用:在庫費用] (生産量 在庫量)

名称: 期	1	2	3
time	240	240	240
Boggie Assembly	[0: 5: 2: 1: 1](0, 0)	[0: 5: 2: 1: 1](0, 0)	[0: 5: 2: 1: 1](0, 0)
Brake Group	[0: 6: 2: 1: 1](0, 0)	[0: 6: 2: 1: 1](0, 0)	[0: 6: 2: 1: 1](0, 0)
Case	[0: 4: 1: 1: 1](0, 0)	[0: 4: 1: 1: 1](0, 0)	[0: 4: 1: 1: 1](0, 0)
case&frame	[0: 5: 3: 5: 6](0, 0)	[0: 5: 3: 5: 6](0, 0)	[0: 5: 3: 5: 6](0, 0)
chassis/platform	[0: 5: 2: 6: 10](0, 0)	[0: 5: 2: 6: 10](0, 0)	[0: 5: 2: 6: 10](0, 0)
common subassembly	[0: 1: 3: 7: 10](0, 0)	[0: 1: 3: 7: 10](0, 0)	[0: 1: 3: 7: 10](0, 0)
Dressed-Out Engine	[0: 7: 6: 6: 9](0, 0)	[0: 7: 6: 6: 9](0, 0)	[0: 7: 6: 6: 9](0, 0)
Drive Group	[0: 3: 1: 1: 1](0, 0)	[0: 3: 1: 1: 1](0, 0)	[0: 3: 1: 1: 1](0, 0)
Engine	[20: 9: 7: 3: 4](0, 0)	[15: 9: 7: 3: 4](0, 0)	[10: 9: 7: 3: 4](0, 0)

ページが表示されました インターネット

BMI固有値問題



SDPA

The screenshot shows a web browser window with the address bar displaying `http://sdparar.dendai.ac.jp/portal/`. The page header features the SDPA Online logo with the tagline "for your future" and the main title "SDPA Online Solver". A binary sequence "0011101111001011001010010101110" is visible on the right side of the header. The navigation bar includes "olver !!", "Pre-Open SDPA Online Solver !", and links for "Top | Support | About us".

The main content area is divided into a left sidebar and a main right section. The sidebar contains three menu sections: "Login Page" with links for "Login", "Login Page", and "Administrator"; "Documents" with links for "User's Manual", "Overview", "Manual", "Cluster Spec.", "Support", "Member", "Link", and "Link"; and "Latest News" with a list of news items including "Pre-Open SDPA Online Solver", "Link", "Administrator Page", "PBS", "Cluster and Member Page", "Three Solvers are available", "Top Page Release", and "Online Solver development".

The main right section features a "Welcome to SDPA Online Solver Home Page" header, followed by a "Latest News" section. The news items are:

- 2005.10.20 **Pre-Open SDPA Online Solver**
[Pre-Open SDPA Online Solver !](#)
- 2005.10.19 **Link**
Link was added in the Documents. [SDPA Online Homepage.](#)
- 2005.09.08 **Administrator Page**
[Administrator Page](#) is opened.
For administrators only.
- 2005.09.07 **PBS (Portable Batch System)**
We have adopted the **Portable Batch System (PBS)** which is flexible batch queuing and workload management system originally developed by Veridian System for NASA.
- 2005.09.03 [Cluster Spec. Page](#) and [Member Page](#) are opened.

SDP (SemiDefinite Programming)

- 半正定値計画問題(最適化問題)

SDP (SemiDefinite Programming)

- オペレーションズリサーチやシステムと制御, 金融工学, 量子化学等幅広い分野への応用

- SDPを解く為のソフトウェア

- SDPA, SDPARA, etc...

- [SDPARA : SemiDefinite Programming Algorithm paRAllel version],

Makoto Yamashita, Katsuki Fujisawa,

and Masakazu Kojima,

Parallel Computing **29**, 1053-1067, 2003.

Webアプリケーションへのログイン

ファイル(E) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)

http://sdparar.dendai.ac.jp/portal/index.php

Welcome login

Please Sign In !

*Login ID :

*Password :

Login Clear

Fields marked with * are required.

[Register](#) (it's free!)

[Return](#)

SDPA Online Solver Main Page

SDPA Online Solver

[User's Manual Page](#)

[Your Download Page](#)

[Parameter file delete](#)

[Data file delete](#)

UserName:

osada

Upload **Parameter** File :

Upload **Data** File :

Select Parameter File :

Select Data File :

Select Solver :

Select Cluster :

Select CPU Number :

-----Optional-----

(Default filename result.out)

Input result filename :

-----Optional-----

(If you need to be notified that the computation finished)

E-mail address :

Mailによる結果通知と情報

Dear osada

Thank you for using SDPA Online Solver !

[source: Katsuki Fujisawa, TDU]

--*-*-*User Information-*-*-*-*-*

Result : result.out

Parameter : param.sdpa

Data : mcp.dat-s

Solver : SDPARA

CPU : 4

--*-*-*

Please download or view result files from our Web site at:

<<<-----Access URL----->>>

[http://sdpara.r.dendai.ac.jp/userpage/\[your LOGIN ID\]/downloads/download.php](http://sdpara.r.dendai.ac.jp/userpage/[your LOGIN ID]/downloads/download.php)

If you have any question, please contact with support@rnc.r.dendai.ac.jp

Sincerely,

SDPA Online Solver Support

Tokyo Denki University
Departments of mathematical Sciences,
Ishizaka, Hatoyama, Hiki, Saitama
〒355-0394, Japan
E-Mail : support@rnc.r.dendai.ac.jp
Fax : +81-492-96-7072

結果

ファイル(F) 編集(E) 表示(V) 移動(G) ブックマーク(B) ツール(T) ヘルプ(H)

All file (Download)					
	File name	Parameter	Data	Last update	Size(KByte)
1	result.out	param.sdpa	mcp.dat-s	December 12 2005 16:04:22.	4.322

Click the file name to download/view the results.

If the result file does not appear here, please "reload" this page after your computation is completed (or receiving the e-mail notification).

If you want to remove unnecessary files, please click. [\[Remove \]](#)

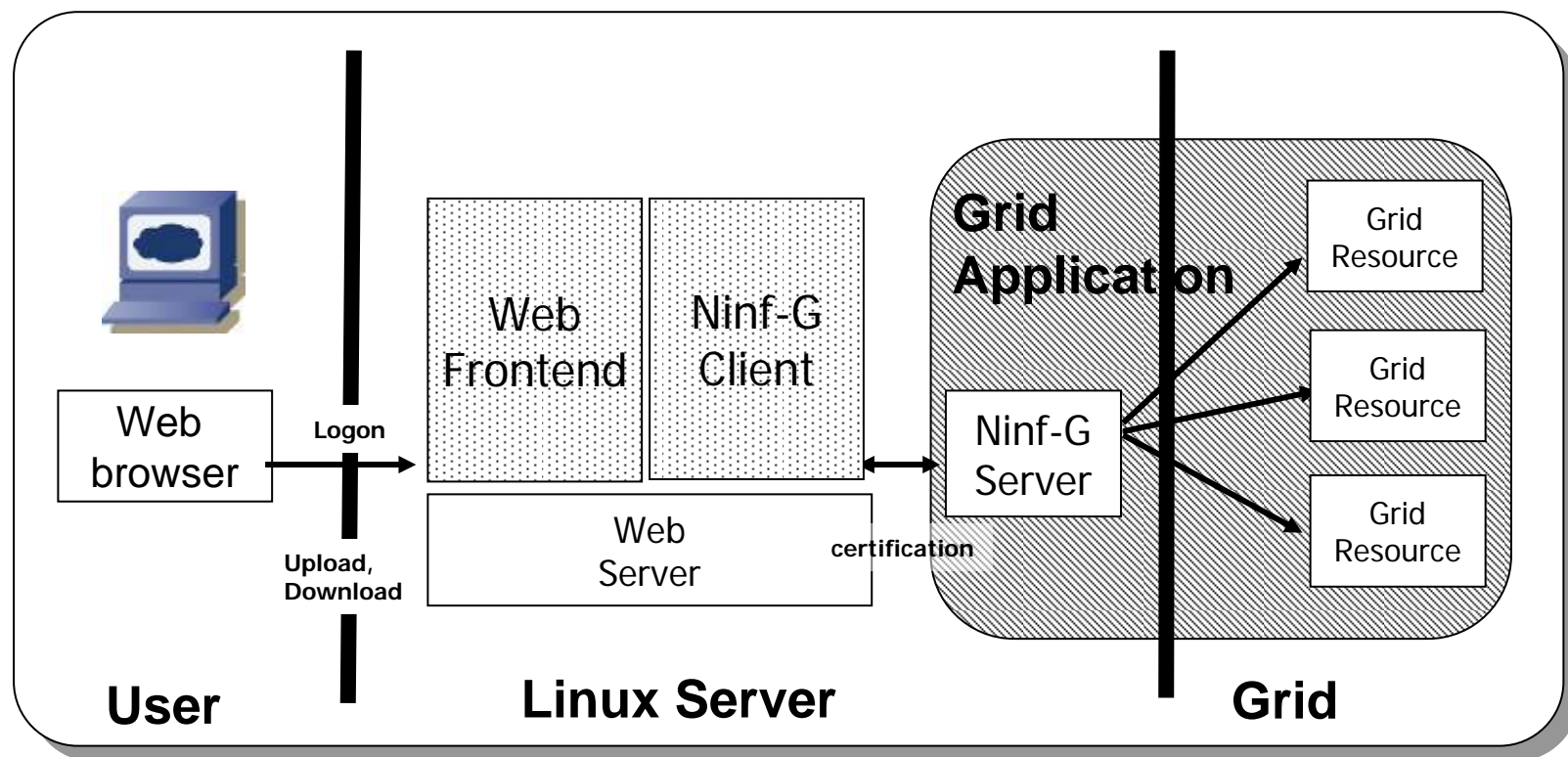
December 12, 2005, 4:04 pm now
File = 1 Total file size=0.004322MB

[Online Solver Main Page Return](#)

logout

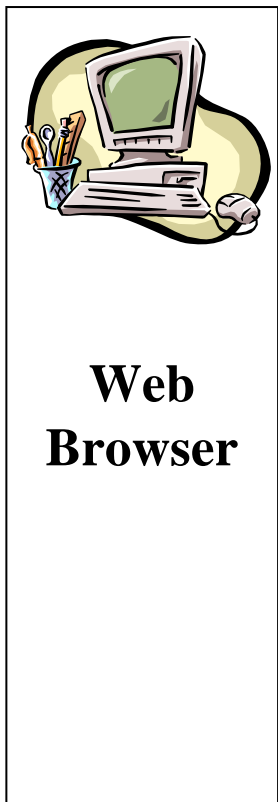
Webアプリケーションの概念図

- ▶ 3層C/Sシステム
- ▶ 簡単にグリッドコンピューティングを実現
- ▶ ブラウザからログインして自分のジョブの実行を依頼



MySQL (Database)

User Data



Web Browser

User

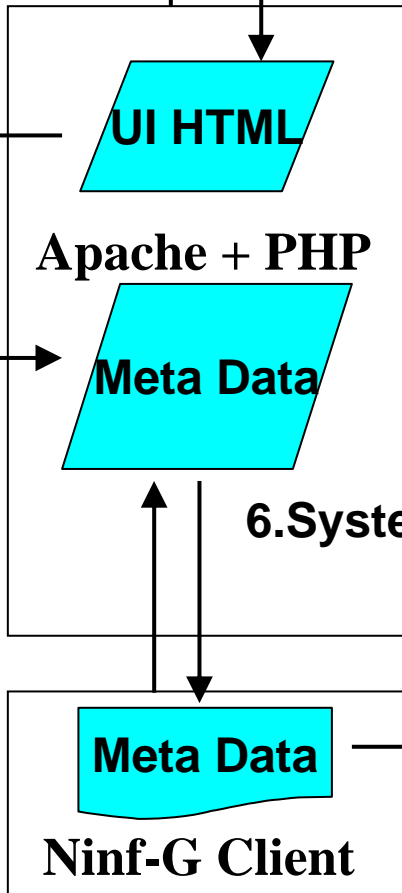
2. Authentication

1. Access

4. Display

5. Submit

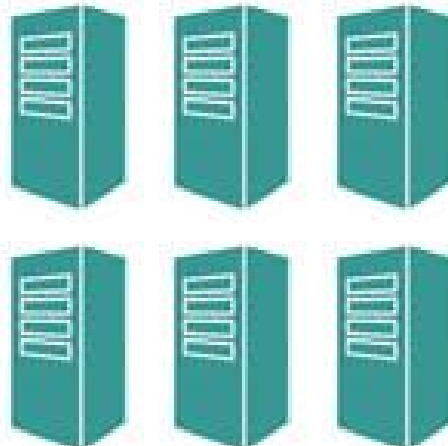
8. Results



Web Server

Job manager (PBS)

Grid Resources



Ninf-1 & MPI

Ninf-G Server

Output

Grid

7. Start up

6. SystemCall

おわりに

- グリッド技術の最適化問題計算への応用に対する期待は大きい.
 - 生命科学, OR, 制御工学, ...
- さらなる発展／普及に向けて, 分野間の連携が重要.
 - アプリケーション分野
 - コンピュータサイエンス分野
 - ネットワーク分野

最適化問題求解アプリケーションの並列化

■ アプリケーションの特徴

- 実行時間の比較的小さいタスクが大量に生成される。
- タスク間の依存関係は(ほとんど)無い。
- 生成されるタスクを計算資源へ次々に割り当てて計算を進めたい。



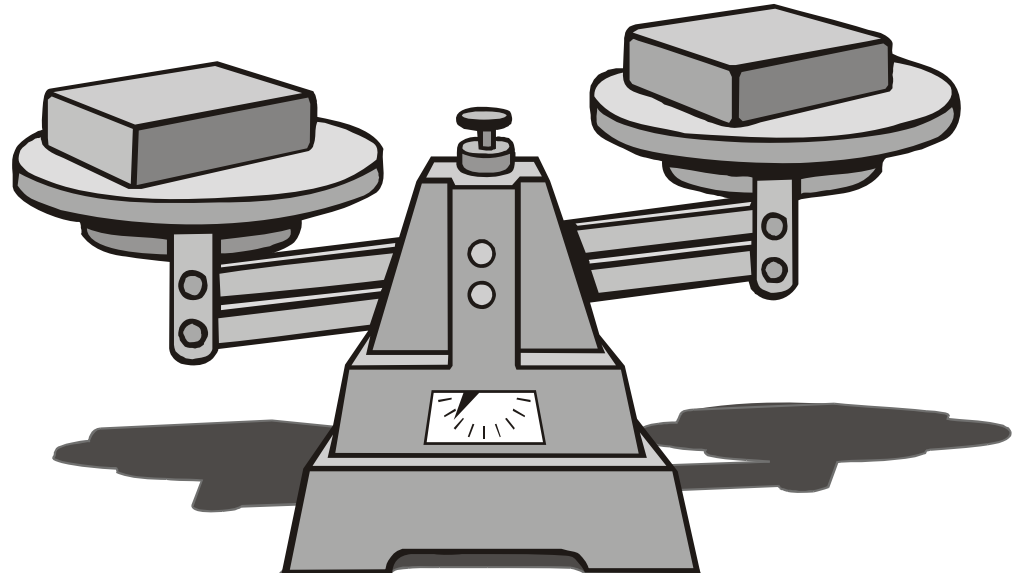
多くの最適化問題求解アプリケーションが
マスタ・ワーカ方式により実装されている。

ロットサイズ最適化システム WebLot

- タクティカルレベルの意思決定
- 在庫と段取り費用のトレードオフ(最適ロットサイズの決定)
- 容量制約を考慮した生産資源への品目の割り当ての最適化
- スケジューリング最適化と合わせることによるAPS (Advanced Planning and Scheduling) の求解

段取り費用

在庫費用

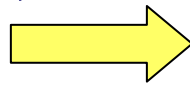


実験(続)

耐障害性の検証

以下のエラー状況を人為的に発生させた

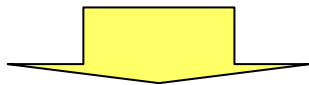
- ・ マスター⇔ゲートウェイ, ゲートウェイ⇔サブマスター, サブマスター⇔ワーカーの通信路が物理的に切断
- ・ マスター⇔ゲートウェイ, ゲートウェイ⇔サブマスター, サブマスター⇔ワーカーのコネクションが切断
- ・ ゲートウェイ, サブマスター, ワーカーのプログラムが異常終了



システム全体の処理は止まらずに正常に計算を続けることを確認

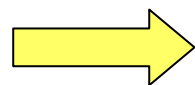
実運用を想定した運用試験

CPU数: 1,196 (5サイト) 打ち切り世代数: 30000 試行回数: 独立3試行の平均



約2時間40分で正常に終了

(Pentium III 1.4GHz / 1CPUで約200日かかる)



実際の規模の問題を現実的な時間内に計算を完了！！

BMI固有値問題

■ 定義

双線形行列関数 $F(x,y)$ の最大固有値を最小化するベクトル変数 x, y を求める問題.

$$F(x,y) := F_{00} + \sum_{i=1}^{n_x} x_i F_{i0} + \sum_{j=1}^{n_y} y_j F_{0j} + \sum_{i=1}^{n_x} \sum_{j=1}^{n_y} x_i y_j F_{ij}$$

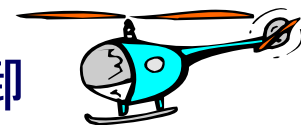
$$F: R^{n_x} \times R^{n_y} \rightarrow R^{m \times m}$$

$$F_{ij} = F_{ij}^T \in R^{m \times m} (i=0, \dots, n_x, j=0, \dots, n_y)$$

■ 応用

■ 制御工学

➤ ヘリコプター制御, ロボットアーム制御



■ OR

➤ 大規模問題求解への挑戦