本資料について

- **本資料は下記書籍を基にして作成されたものです。文章の内容の正確さは保障できないため、正確な知識を求める方は原文を参照してください。**

  - **著者：田中良夫　平野基考　佐藤三久　中田秀基　関口智嗣**
  - **論文名：ファイアウォールに対応したGlobusによる広域クラスタシステムの構築とその評価**
  - **出展：情報処理学会論文誌** Vol. 41 No.SIG8
  - **発表日：**2000**年**8**月**

ファイアウォールに対応したGlobusによる広域クラスタシステムの構築とその評価

原文

田中良夫　平野基考　佐藤三久　中田秀基　関口智嗣

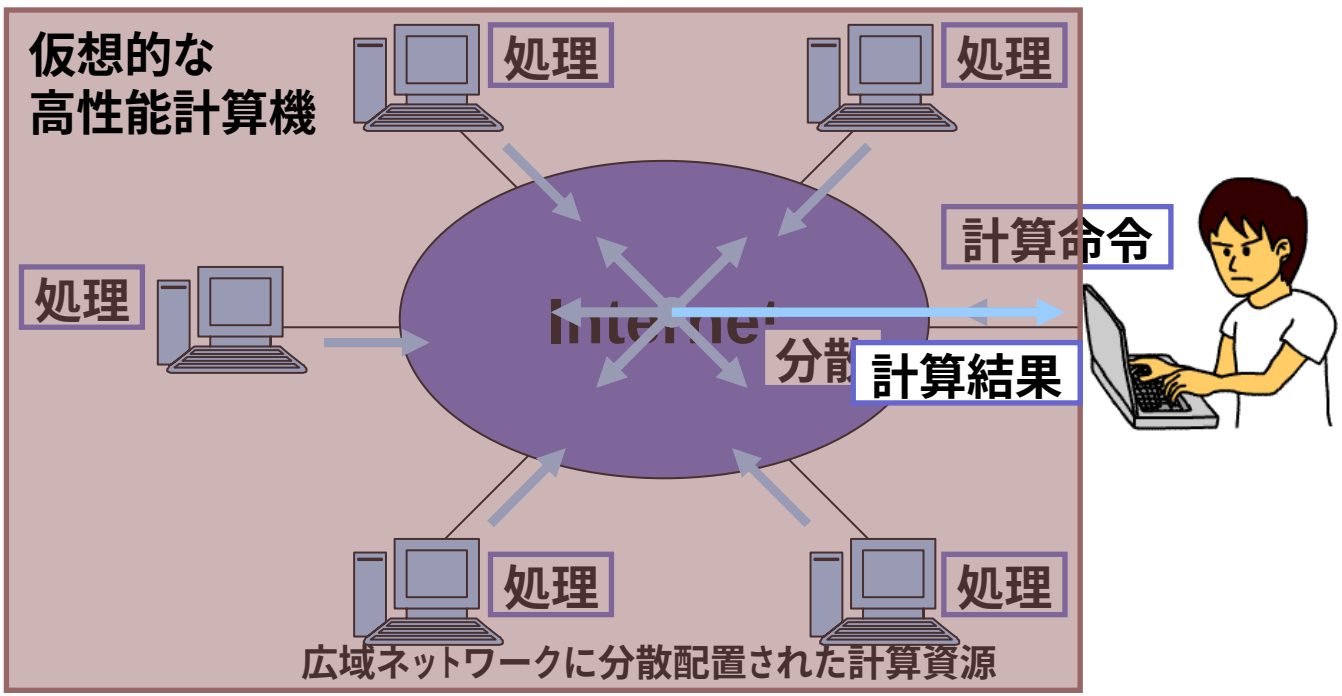
発表

渡邊研究室　伊藤将志

1．はじめに

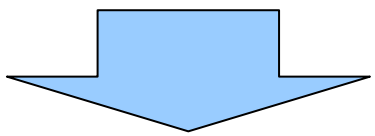
- **グローバルコンピューティング**

  近年、広域に配置された計算資源を利用するグローバルコンピューティングに関する研究がさかんである



グローバルコンピューティング

- 必要な要素技術
  - ユーザ認証
  - 通信
  - 遠隔計算機でのプロセス生成



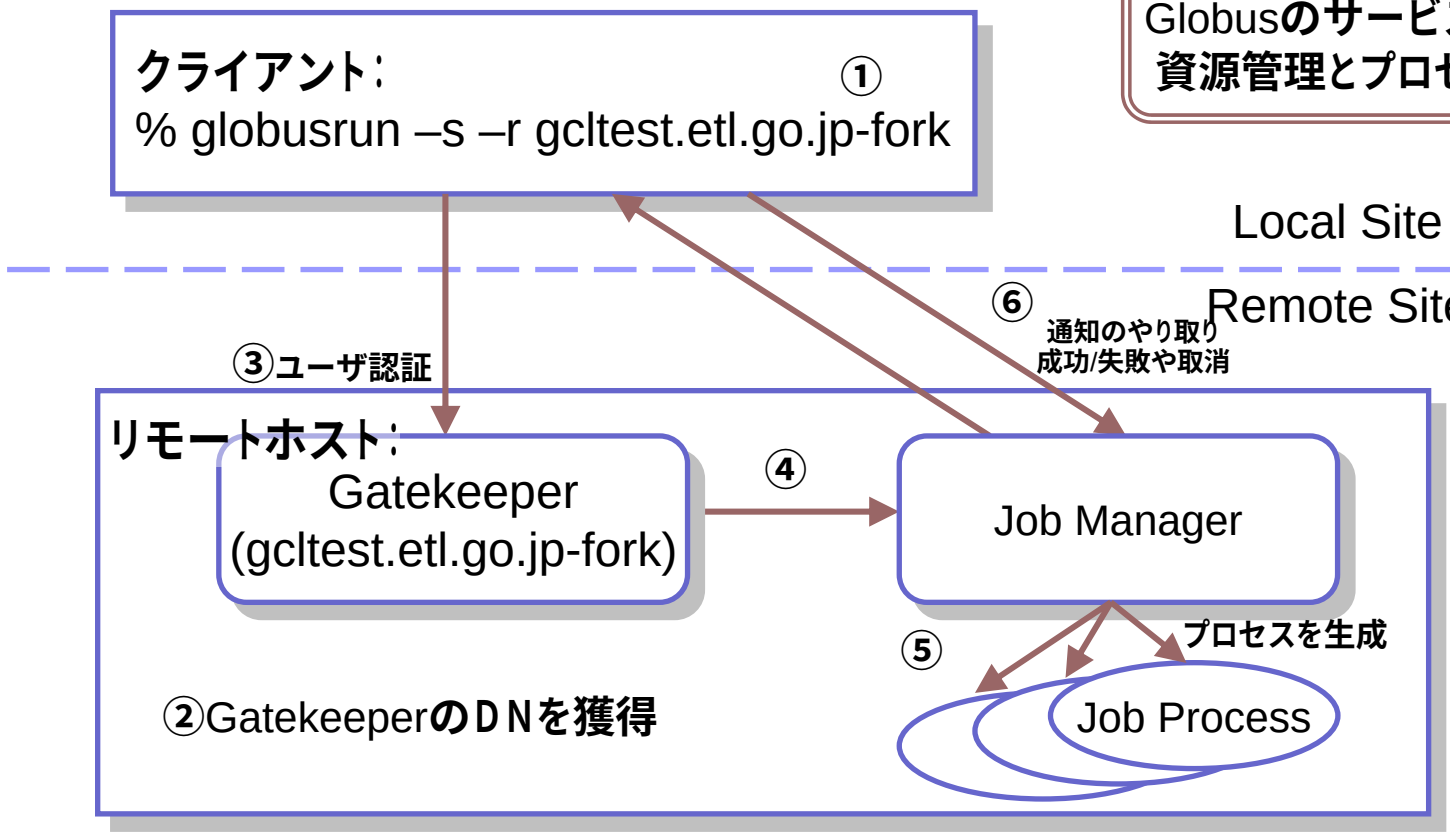
Globusが提供

- 米国の大規模な研究チームによる低レベルツールキッド
- ソフトウェアインフラストラクチャを構成する事実上の標準
- 欠点が多いので改良が必要

Globus

- **Globusの仕組み**

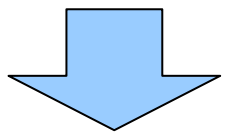
GRAM
Globusのサービスの一つ
資源管理とプロセス生成



クライアントからリモートホストに送信されたジョブが実行される様子

技術の拡張

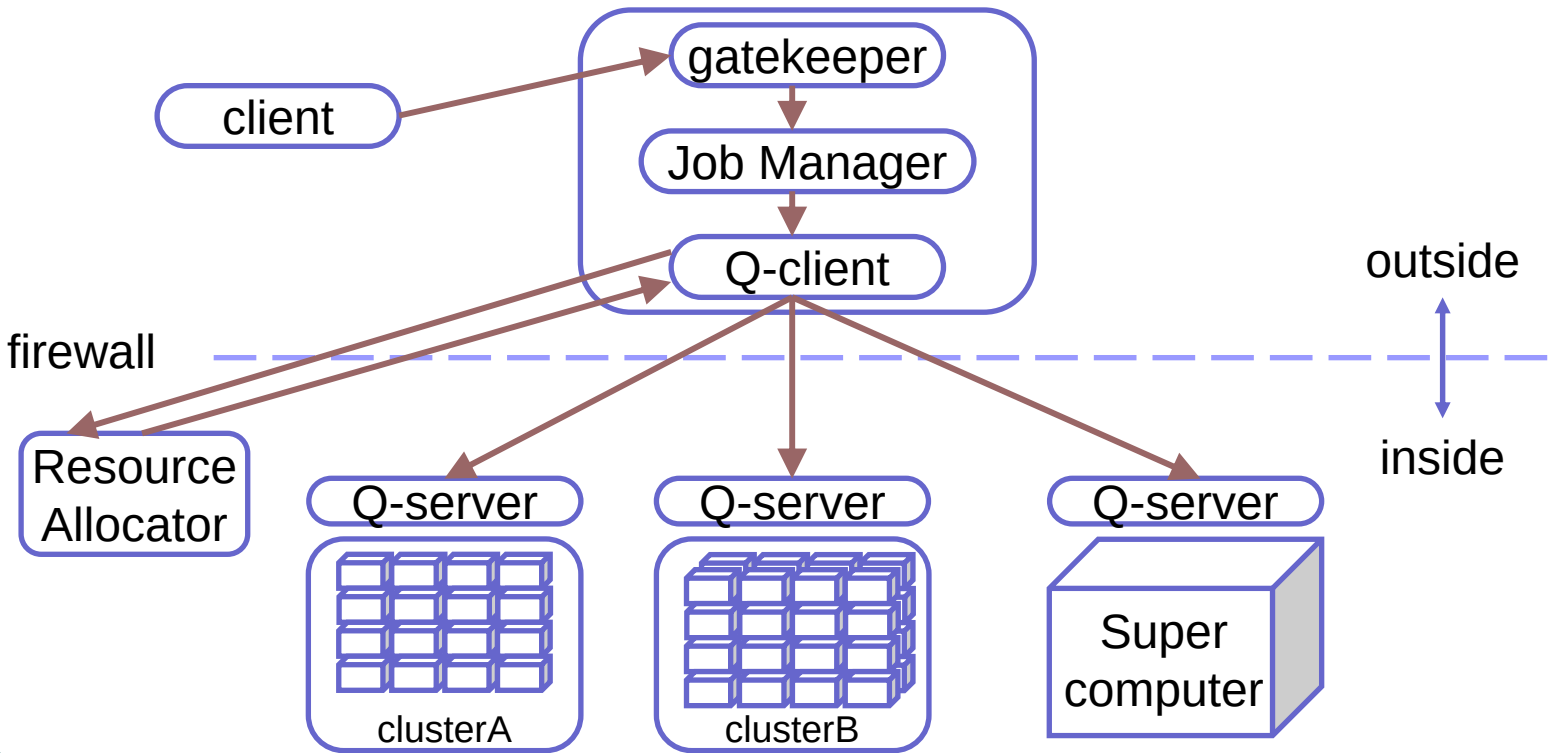
- Globusの問題点
  - クラスタシステムの使用に向いていない
    - すべての計算機にGlobusをインストールしなければならない
    - ＬＳＦ（リソースマネージャ）ではクラスタを一台の仮想的な並列計算機として扱えない
  - ファイアウォールを超えた処理ができない
    - 通信の際Globusは動的にポートを決定してしまう



- 解決策
  - クラスタシステムの使用
    →RMF（リソースマネージャー）
  - ファイアウォールの通過
    →Nexus Proxy

RMF

- ファイアウォール内にある資源を利用するためのシステム

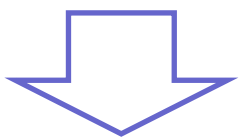


RMF

計算資源コンフィグレーションファイルの例

#NAME	Type	Procs	Nodes	Clock	Prefix
COMPaS	C	4	8	200	Compas
COMPaS2	C	4	4	450	Compas-2
SR2201	P	1	256	150	sr2201

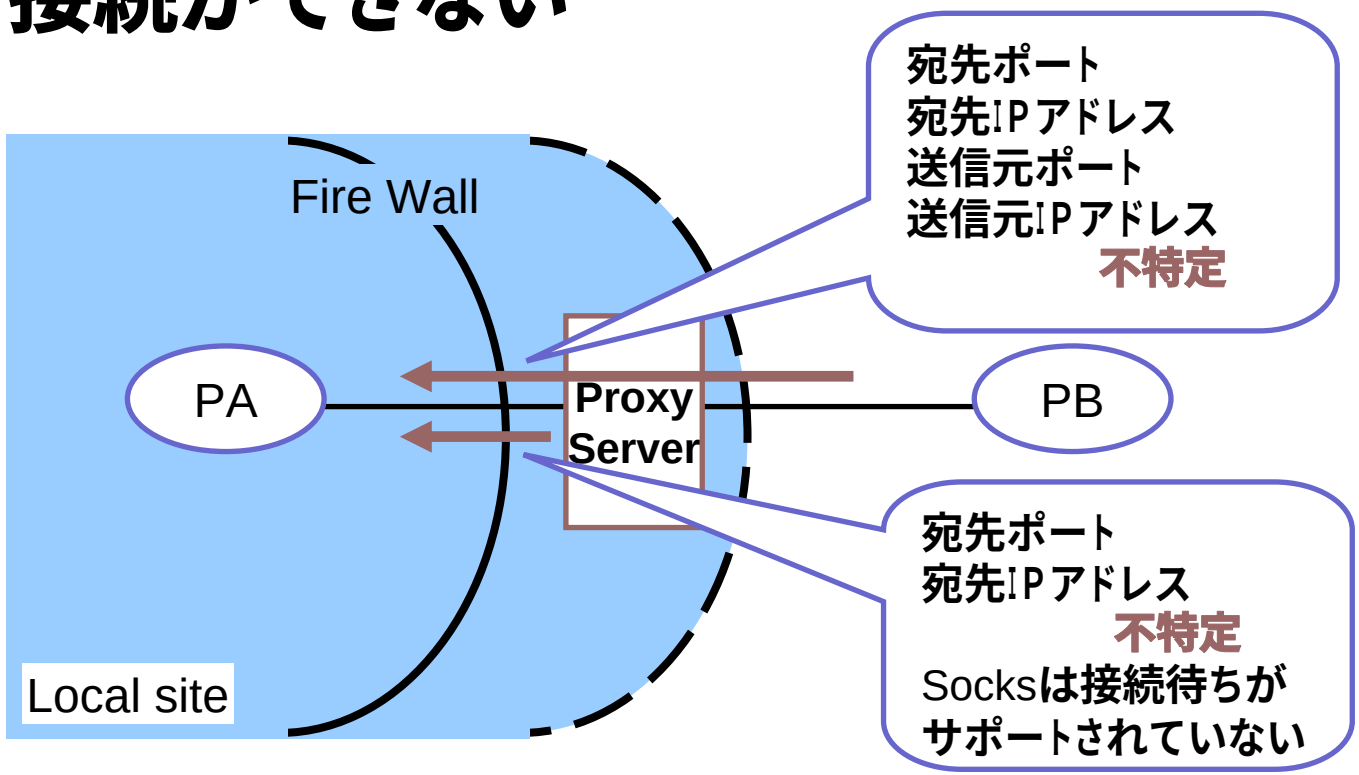
Compasの8台のノード名
Compas0
Compas1
:
compas7



クラスタを一台の並列計算機と仮想できる

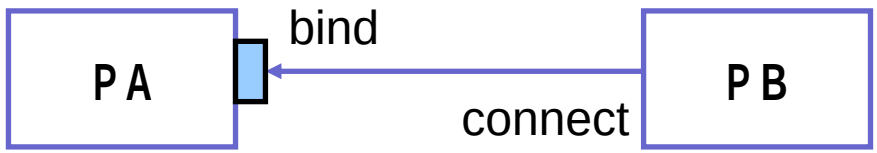
ファイアウォールの問題

- ファイアウォールのため外部から内部への接続ができない



Initial passive socket open

- Globusには被接続ポイントを先に確立し、発呼側からの接続を待つ機能が必要



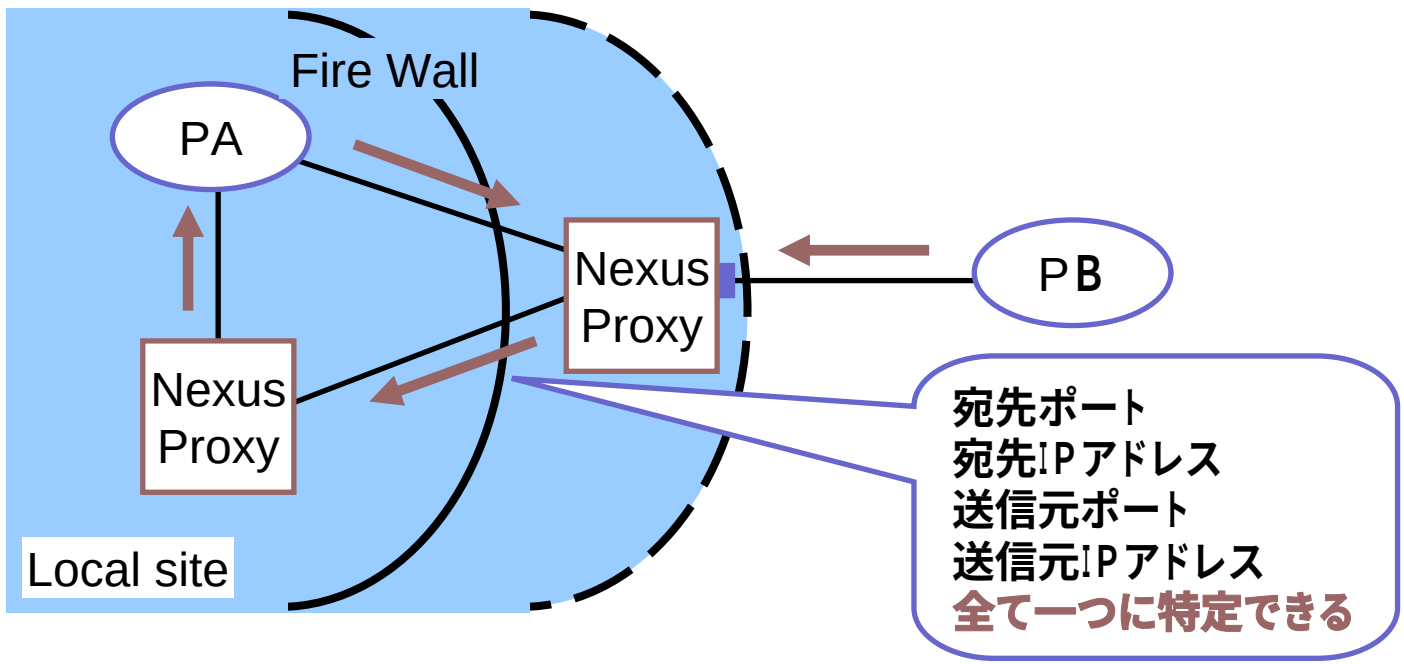
SOCKSプロトコルではサポートされていない機能

Nexus Proxyによって可能

Nexus Proxyの設計と実装

- Nexus Proxy

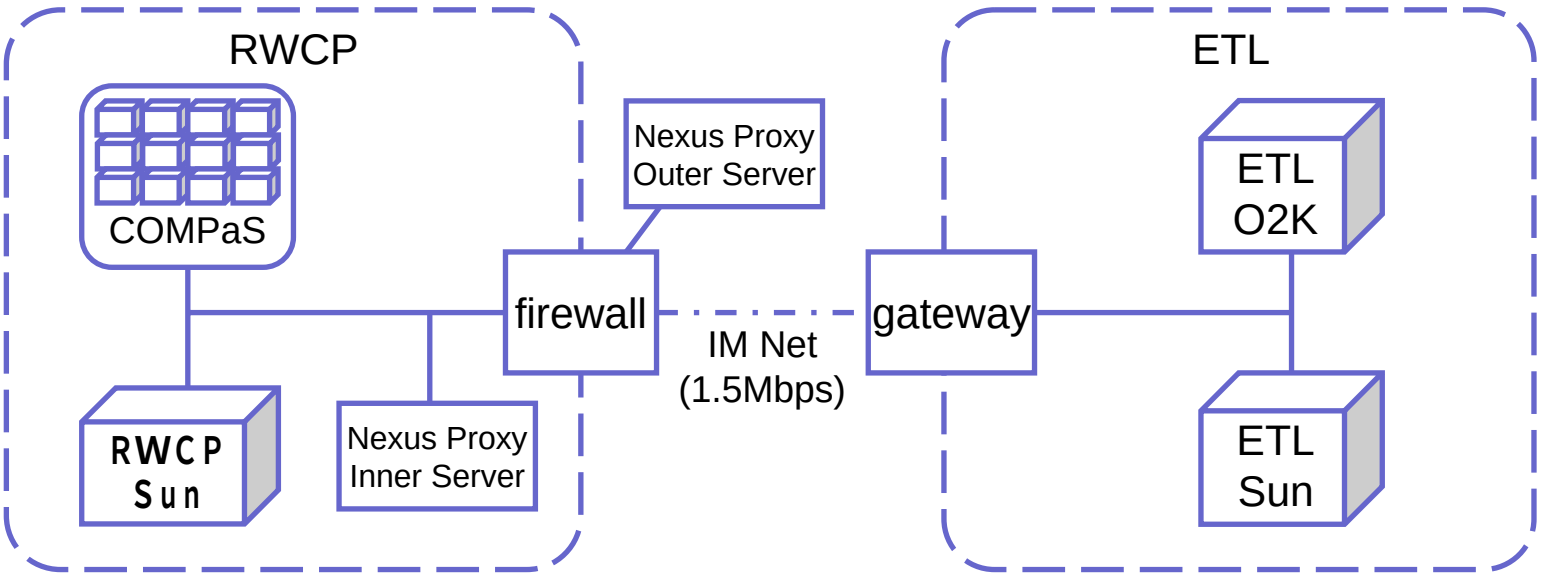
  外部から内部に接続するためのメカニズム



実験

- **実験環境**

  RMFとNexusProxyを組み込んだGlobusを用いて、以下の実験環境で実験を行う



実験1

- Nexus Proxyの性能

実験

- RWCP-SunとETL-Sunの間で通信遅延、バンド幅を測定
- Nexus Proxyを介した場合、直接通信した場合を比較する

	遅延 (0B message)	バンド幅(4KB message)	バンド幅(1MB message)
RWCP-Sun ⇔ETL-Sun(direct)	3.9msec	112.0KB/sec	174.4KB/sec
RWCP-Sun ⇔ETL-Sun(indirect)	25.1msec	109.5KB/sec	178.1KB/sec

メッセージを中継する際、ＴＣＰストリームの
コピー処理がユーザプロセスにより行われるため

結果

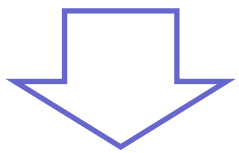
- Nexus Proxyを介した場合と直接では遅延は約6倍
- メッセージサイズが大きくなるとデータ転送時間が増加し、コピー処理によるオーバーヘッドが無視できる

実験2

- 分岐限定法によるナップサック問題の実装

  広域並列システムでは次のようなアプリケーションが高い効率を得られる可能性がある

  - 各プロセッサが非同期に計算を進められる
  - データの独立性が高い
  - 計算量が多く、高い並列性を持つ



木探索が最適

ナップサック問題を分枝限定法によって並列に解くプログラムを実装

実験2

名称	概要
COMPaS	COMPaSの8プロセッサ.全部で8ノード,1ノードにつき1プロセッサを利用.
ETL-O2K	ETL-O2Kの8プロセッサ.
Local-area Cluster	RWCP-Sun + COMPaS.全部で12プロセッサ.RWCP-Sunの4プロセッサ,COMPaSの8プロセッサを利用.
Wide-area Cluster	RWCP-Sun + COMPas + ETL-O2K.全部で20プ…の4…8プ…の8プロセッサを利用.

実験に用いたテ…

システム	プロセッサ数	実行時間(sec)	速度向上率
RWCP-Sun	1	26547	1
COMPaS	1	23211	1.14
COMPaS	8	3135	8.47
ETL-O2K	8	6849	3.88
Local-area Cluster	12	2936	9.04
Wide-area Cluster(use Nexus Proxy)	20	2074	12.80
Wide-area Cluster(direct communication)	20	2003	13.25

並列化効率 約66%

並列化効率 約64%

12プロセッサで9倍、約75%の並列化効率

RWCP-SunとCOMPaSの処理能力は同等と見れる

ナップサック問題の実行時間

- RWCP-SunとCOMPaSの間もNexus Proxyを介しているため実行時間の差が小さい
- ETL-O2Kの性能がRWCP-SunやCOMPaSに比べ劣り、RWCPとETL間のネットワーク性能が低い
- 広域環境ではProxyを介した通信を行うことによるオーバヘッドは無視できる

実験2

System	Master	RWCP-Sun			COMPaS			ETL-O2K		
		Max	Min	Average	Max	Min	Average	Max	Min	Average
Local-areaCluster	160459	13869	15649	14981	17219	11385	14436			
Wide-area Cluster	217330	11603	8394	10563	13289	8007	11465	8508	2105	5693

親ノードが処理したジョブ要求の回数

RWCP-SunとCOMPaS上の子ノードは約0.2秒に一度、ETL-O2Kは0.36秒に一度の割合でジョブ要求を出している

⇒ ジョブが粗粒度なので通信量が増加する傾向

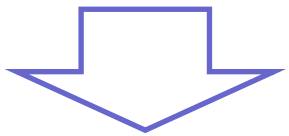
System	Master	RWCP-Sun			COMPaS			ETL-O2K		
		Max	Min	Average	Max	Min	Average	Max	Min	Average
Local-areaCluster	26.6	45.6	44.4	44.8	47.0	43.4	45.0			
Wide-area Cluster	14.7	34.3	31.7	32.7	34.9	31.0	32.5	20.3	17.4	18.5

走査したノード数（単位は億）

非均質な環境でも計算機の性能に応じて効果的な負荷分散が行われている

まとめ

- Globusにおいてファイアウォールを超えたクラスタや並列計算機などの利用



新しい形のＧＲＡＭであるＲＭＦ
Ｎｅｘｕｓ　Ｐｒｏｘｙ

を設計

- 実験の結果、効果的な付加分散、通信料の抑制や通信と計算のオーバラップなどを意識したプログラミングを動かした場合、広域クラスタシステムでも十分に受け入られる性能をもつ

Thank you for your attention

Masashi Ito

ファイアウォールに対応したGlobusによる広域クラスタシステムの構築とその評価