

広域グリッドコンピューティングのための 制御理論にもとづく動的資源管理方式

大阪大学
大学院情報科学研究科
情報ネットワーク学専攻

大崎 博之

1

発表の内容

- はじめに
- グリッドコンピューティングにおける資源管理
- フィードバック制御としてのモデル化
- 動的資源管理方式 DRM-DC
- シミュレーションによる性能評価
- まとめと今後の課題

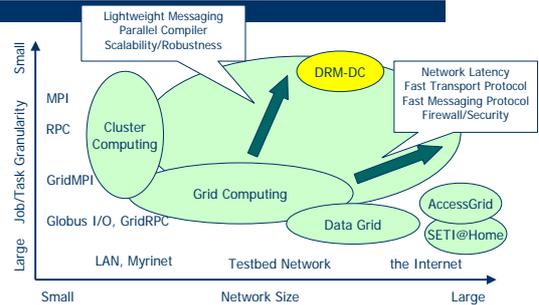
2

研究の背景

- グリッドコンピューティングの登場
 - 分散する計算機資源を統合的に利用
 - 大規模な科学技術計算を可能に
 - GGF(Global Grid Forum)における標準化
 - 広域グリッドコンピューティングにおける問題点
 - サイトの利用可能資源量が時間とともに変動
 - 資源が複数の利用者により共有
 - 例: 計算機資源、ネットワーク資源、ディスク資源
 - ネットワークの伝搬遅延が無視できない
 - 広域ネットワークではラウンドトリップ時間 > 100ms
- 動的な資源管理方式が必要

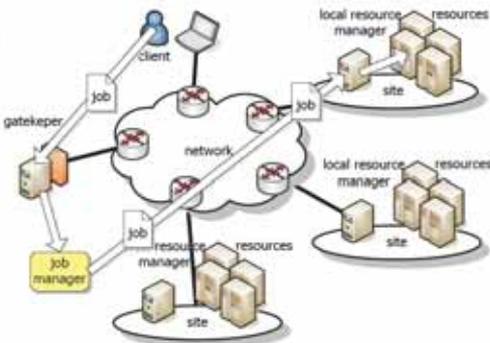
3

グリッドコンピューティングの現状と課題



4

グリッドコンピューティングにおける資源管理



5

GRAM (Grid Resource Allocation and Management)

- Globus ツールキットの資源管理コンポーネント
- GRAM の構成要素
 - クライアント
 - ジョブ生成
 - ゲートキーパ
 - クライアント認証、ジョブマネージャ生成
 - ジョブマネージャ
 - ローカル資源マネージャへジョブ割り当て、ジョブ実行状態の監視
 - ローカル資源マネージャ
 - ジョブをサイト内の資源で実行、結果をジョブマネージャに通知
 - サイト
 - 仮想組織に属する資源の集合

6

グリッドコンピューティングにおける資源管理

- 資源管理の目的
 - サイトへ投入するジョブ量を動的に調整
 - 時間とともに変動する各サイトの利用可能資源量にあわせる
- 資源管理の特性
 - 定常特性
 - サイトの資源が高い利用率を維持できること
 - ジョブの実行が完了するまでの時間が短いこと
 - 過渡特性
 - 立ち上がり時間といった系が安定するまでのふるまい
 - 安定性
 - ある程度の時間が経過すればシステムが安定するかどうかが
 - 堅牢性
 - 障害が発生したとしても正常に動作するか
 - 柔軟性
 - さまざまなパラメータ条件下でも良好に動作すること

7

広域グリッドコンピューティングにおける資源管理の問題点

- サイトの利用可能資源量が時間とともに変動
 - 単純な閉ループ型の制御では不十分
 - フィードバック情報を利用した閉ループ型の制御が不可欠
- ネットワークの伝搬遅延が無視できない
 - 単純なフィードバック制御では安定性や堅牢性を実現できない
 - フィードバック遅延を考慮した制御が必要

8

研究の目的

- 動的資源管理方式 DRM-DC の提案
 - DRM-DC (Dynamic Resource Management with Delay Compensator)
 - フィードバック制御を利用
 - 遅延補償器(スミス予測器)を利用
 - 高い定常特性と過渡特性を実現
- 動的資源管理方式 DRM-DC の性能評価
 - 離散時間シミュレータSimgridに DRM-DC を実装
 - 提案するDRM-DCとPI制御器を比較
 - 広域グリッドコンピューティングにおける有効性を示す

9

モデル化における仮定

- パラメータ探索型のアプリケーション
 - ジョブマネージャは常に実行すべきジョブ(複数のタスクにより構成)を持つ
 - タスクの粒度はジョブマネージャ-サイト間の転送遅延と比較して十分小さい
- サイトの処理能力
 - サイトの処理能力はローカル資源マネージャのバッファ内タスク数によらず一定
- ネットワークの遅延
 - ジョブマネージャ-サイト間の伝搬遅延は既知であり、時間とともに変動しない

10

フィードバック制御としてのモデル化: (1) サイトのモデル化

- 単一の待ち行列としてモデル化
 - サイトは割り当てられた一連のタスクを順次処理
 1. ジョブマネージャからタスクが投入される
 2. タスクをいったんローカル資源マネージャのバッファに格納
 3. 利用可能資源量に応じてタスクを順次実行
- 連続時間系の制御システムとしてモデル化
 - サイトは制御対象(プラント)に相当
 - 入力 $u(t)$: ジョブマネージャからサイトへのタスク投入レート
 - 出力 $x(t)$: ローカル資源マネージャのバッファ内タスク数
 - 積分器によってモデル化が可能

$$x(t) = \left[\int_0^t (u(v) - \tau(v)) - \mu(v) dv \right]^+$$

11

フィードバック制御としてのモデル化: (2) ジョブマネージャのモデル化

- 制御器としてモデル化
 - サイトへのタスク投入量を調整
 1. ジョブマネージャがサイトからのフィードバック情報を受信
 2. サイトへのタスク投入量を調整
- 連続時間系の制御システムとしてモデル化
 - ジョブマネージャは制御器(コントローラ)に相当
 - 入力 $x(t)$: サイトからのバッファ内タスク数
 - 出力 $u(t)$: サイトへのタスク投入レート

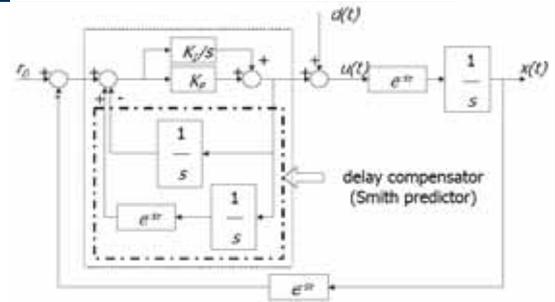
12

動的資源管理方式 DRM-DC

- DRM-DC (Dynamic Resource Management with Delay Compensator)
- ジョブマネージャ上で動作する制御器(コントローラ)
 - ローカル資源マネージャのバッファ内タスク数がフィードバック情報
 - ローカル資源マネージャへのタスク投入レートを制御
- それぞれのサイトごとに独立にフィードバック型の制御を行う
 - 一般には、ジョブマネージャが管理するサイトは複数存在
- DRM-DC の制御目標
 - ローカル資源マネージャのキュー長を一定値に保つこと
 - サイト内資源が過負荷になることを防ぐ
 - サイト内資源の利用率の向上を図る
 - タスク実行完了までの待ち時間を小さく
- 遅延補償器を用いたPI(Proportional Integral)制御

13

動的資源管理方式 DRM-DC のブロック図



14

シミュレーションによる性能評価

- グリッド用の離散時間シミュレータSimgridを修正して使用
 - パラメータ探索型のアプリケーションを想定
 - 一定の制御間隔Tごとに制御を行う離散時間型のDRM-DCを実装
- シミュレーション条件
 - ジョブマネージャが管理するサイト数: 10
 - ネットワーク環境: LAN (1[ms]) および WAN (100[ms])
 - シミュレーション時間: 15 秒
- シミュレーション実験におけるDRM-DCの性能指標
 - ローカル資源マネージャのキュー長(バッファ内タスク数)
 - 時間的変動、立ち上がり時間、行き過ぎ量、整定時間

15

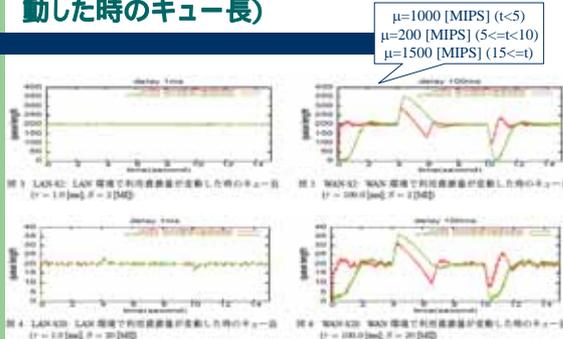
シナリオ 1 のパラメータ設定

- シミュレーションシナリオ
 1. サイトの利用可能資源量が時間とともに変動する場合
 2. ジョブマネージャ-サイト間の遅延が時間とともに変動する場合

	LAN-S2	LAN-S20	WAN-S2	WAN-S20
タスクサイズ S [M]	2	20	2	20
転送遅延 τ [ms]	1.0	1.0	100.0	100.0
制御間隔 T [ms]	2.0	20.0	2.0	20.0
目標キュー長 r_0	200	20	200	20

16

シミュレーション結果 (利用可能資源量が変動した時のキュー長)



17

シミュレーション結果 (立ち上がり時間、行き過ぎ量、整定時間)

- 立ち上がり時間
 - キュー長が目標値の10%から90%に遷移するまでの時間
- 行き過ぎ量
 - キュー長の最大値が目標値をどれだけ上回ったか
- 整定時間
 - キュー長が目標値の5%以内安定するまでの時間

転送遅延 τ [ms]	タスクサイズ S [M]	立ち上がり時間 [s]	行き過ぎ量 [%]	整定時間 [s]		
LAN-S2	DRM-DC	1.0	2	0.050	0.3	0.012
LAN-S2	PI	1.0	2	0.010	0.3	0.008
LAN-S20	DRM-DC	1.0	20	0.4	5	0.8
LAN-S20	PI	1.0	20	0.4	5	0.8
WAN-S2	DRM-DC	100.0	2	0.30	48	0.11
WAN-S2	PI	100.0	2	1.6	79	1.7
WAN-S20	DRM-DC	100.0	20	0.10	55	0.3
WAN-S20	PI	100.0	20	1.02	85	1.6

18

まとめと今後の課題

- 動的資源管理方式 DRM-DC を提案
 - 広域グリッドコンピューティングが対象
 - フィードバック制御における遅延補償器(スミス予測器)を利用
 - 高い安定性と過渡特性を実現
- シミュレーション実験による有効性の検証
 - 非常に優れた定常特性および過渡特性を確認
 - 特に転送遅延の大きな WAN 環境において有効
- 今後の課題
 - 安定性、堅牢性、柔軟性の向上を図る
 - 制御パラメータ設定方法、モデル化誤差の減少、サイトモデルの同定
 - Globusツールキットへの組み込み