

複数ドメイン環境でQoSを保証する クラウドのための資源管理フレームワーク

竹房あつ子, 中田秀基, 高野了成, 柳田誠也,
大久保克彦, 工藤知宏, 田中良夫

産業技術総合研究所

クラウドの課題

- クラウド=ハードウェア, プラットフォーム, アプリケーションをインターネットを介してサービスとして提供
- 複数提供者(ドメイン)の資源を組合わせてサービスを提供する際の課題
 - 資源管理フレームワーク
GridARSを開発
 - 性能保証
 - ネットワークも資源として扱い, 性能を保証
 - スケジューリング
 - サービスに必要な性能を満たす資源の組合わせを探索
 - 仮想化
 - 確保された資源を仮想化し, 1つの実行環境として提供
 - モニタリング
 - 確保された資源の利用状況を収集し, ユーザに提供

GridARS資源管理フレームワーク

- GridARSの特徴
 - 予約を前提とし、複数ドメインの多様な資源の性能を保証
 - 資源を適切に組合わせて実行環境を提供
 - 確保した資源のモニタリング情報も提供
- GridARSのコンポーネント
 - 資源管理システム RMS: 性能保証＋スケジューリング
 - 分散モニタリングシステム DMS: モニタリング
 - アプリケーション実行環境管理システム AEM: 実行環境の提供

GridARSの課題

- 資源管理システム RMS
 - G-lambdaプロジェクトで開発されたウェブサービスI/F GNS-WSIを採用
 - GNS-WSI v. 2でGlobus Toolkit 4.0 (GT4)で実装
 - GT4は一般的なWS実装と相互運用性なし
 - 新たにGNS-WSI v. 3 (GNS-WSI3)が規定された
- 分散モニタリングシステム DMS
 - GridARS RMSと連携して動作
 - GT4版のみで、実環境でのRMSとの連携はまだ

GNS-WSI3ベースRMS, DMSのGT4版とPureWS版を開発

- GT4版, PureWS版(Apache CXF+Jetty)の基本性能の評価
- SC09におけるRMS, DMSの連携実証実験

発表概要

- GridARS資源管理フレームワークの概要
- GNS-WSI3ベースGridARSの開発
 - RMS, DMSのGT4版とPureWS版を開発
- GridARSのエミュレーション環境での実験
 - GT4版とPureWS版RMSの基本性能
 - PureWS版DMSの基本性能
 - SC09での実証実験
- 関連研究
- まとめと今後の課題

GridARS資源管理フレームワークの概要

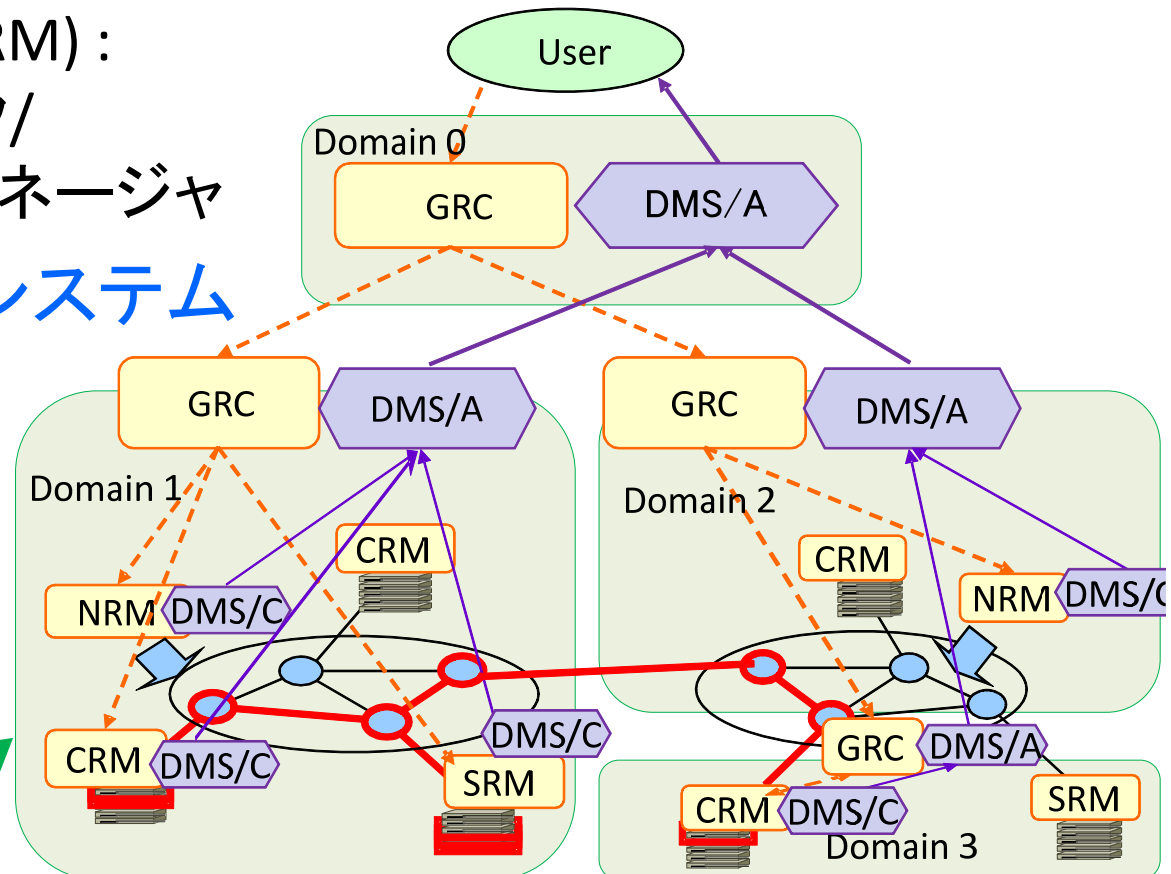
- 資源管理システム RMS

- GRC : グローバル資源コーディネータ
- RM (CRM/NRM/SRM) : (計算/ネットワーク/ストレージ)資源マネージャ

- 分散モニタリングシステム DMS

- DMS/A : アグリゲータ
- DMS/C : コレクタ

Domain : 組織など
資源管理ドメイン



資源管理システム RMS

- RMSの構成

- GRC : グローバル資源コーディネータ

- ユーザの要求を満たす資源を適切に選択
 - RMと連携して選択資源を確保, 提供

- RM : 資源マネージャ

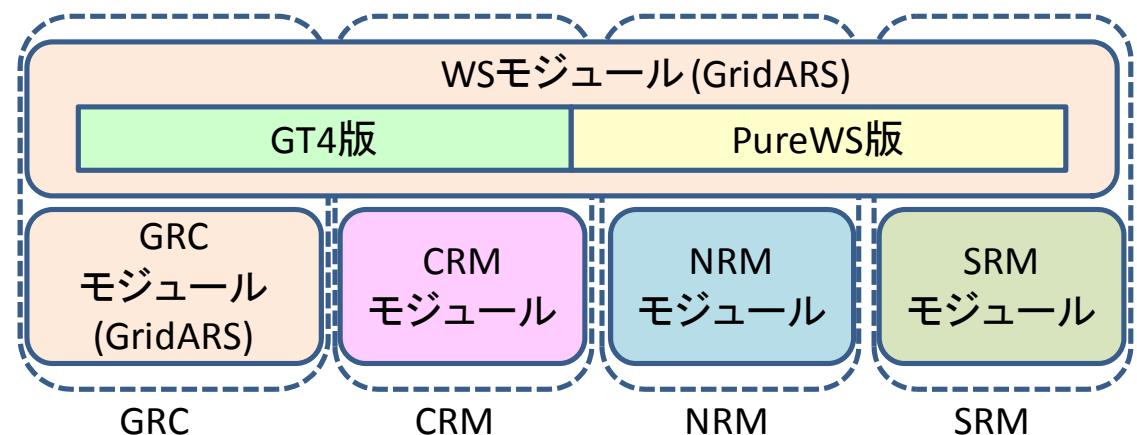
- 各資源を管理
 - バックエンドに予約機能を持つ資源管理システムを利用

- RMSのモジュール

- WSモジュール

- GRC/RMモジュール

- 独自のGRC/RMの実装も可能



分散モニタリングシステム DMS

- DMSの機能
 - RMSで確保された資源群のモニタリング情報を提供
 - モニタリング情報は各ドメインで管理され, 集中DBなし
- DMSの構成
 - DMS/A :アグリゲータ
 - RMSと連携し, 予約ツリーを辿ってモニタリング情報を収集
 - DMS/C :コレクタ
 - 資源のモニタリング (E.g., SNMP, lperf, ping)
 - 各ユーザに対し, どの情報を提供するかは各ドメイン決定 (XACMLを用いた認可)

GNS-WSI3ベースGridARSの開発

- GNS-WSI3インタフェースの特徴
 - G-lambdaプロジェクトで新たに規定
 - 多様な資源の予約手続きのためのWS I/F
 - v. 1, 2はネットワーク資源のみ, v. 3では扱う資源を抽象化
 - WSRFベースで2相コミットによる手続き
 - v. 1はWS, v. 2からWSRF, 2相コミットを採用
 - インタフェースの規定のみ
- GNS-WSI3ベースGridARS RMS, DMSでは
GT4版(GT4.2)とPureWS版(Apache CXF)を開発

RMSインタフェース=GNS-WSI3

オペレーション	機能	入力	出力
create	初期化		RsvEPR
reserve	資源予約 要求	RsvEPR, 時刻, 資源	CmdEPR
modify	修正要求	RsvEPR, 時刻, 資源	CmdEPR
release	解放要求	RsvEPR, 資源ID	CmdEPR
commit	仮手続きの 実行	CmdEPR	
abort	仮手続きの 破棄	CmdEPR	
getResource Property	プロパティ 値の取得	プロパティ 名	プロパ ティ値

- GNS-WSI3オペレーション
 - GRC, RMとも共通
 - ポーリングベース
(ノンブロッキング)
 - 2相コミット
- 予約手続き
 - create (初期化)
 - reserve (仮予約)
 - getCommandStatus
(仮予約状況確認)
 - commit (予約実行)

RsvEPR : 予約手続きID

CmdEPR : 仮手続きID

DMSインタフェース

オペレーション	機能	入力	出力
create	初期化	RsvEPR	MonEPR
configure	収集情報, 時刻の指定	MonEPR, 時刻, 情報リスト	
getInformation	指定した情報の取得	MonEPR (, 時刻)	モニタリング情報

RsvEPR : RMSの予約手続きID

MonEPR : モニタリング手続きID

- DMSオペレーション

- DMS/A,Cとも共通
- 基本的にブロッキング
- configure時に, RMSの予約情報から予約ツリーを辿る

→DMS/Aのconfigureのみ
ノンブロッキング

- モニタリング手続き

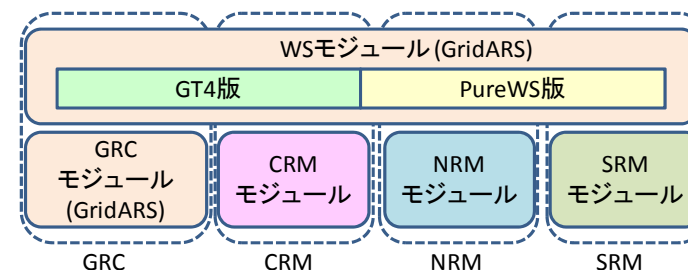
- create (事前)
- configure (事前)
- getInformation (予約時刻)

GT4版とPureWS版の実装

- GT4版
 - Globus Toolkit 4.2を利用
 - WSRF標準やセキュリティ機能(GSI)をサポート
 - 一般的なWS実装とのインターオペラビリティなし
 - バグ修正はGlobusプロジェクト依存
 - マルチスレッドで複数RMへの初回アクセス時に例外発生

- PureWS版
 - Apache CXF 2.2.3(WSコンテナ)とJetty 7.0.0(ウェブサーバ)を利用
 - 一般的なWS実装の一つ
 - WSRF標準やセキュリティ機能なし

- GT4版とPureWS版のWSモジュール及びGRC/RMモジュールをそれぞれ実装



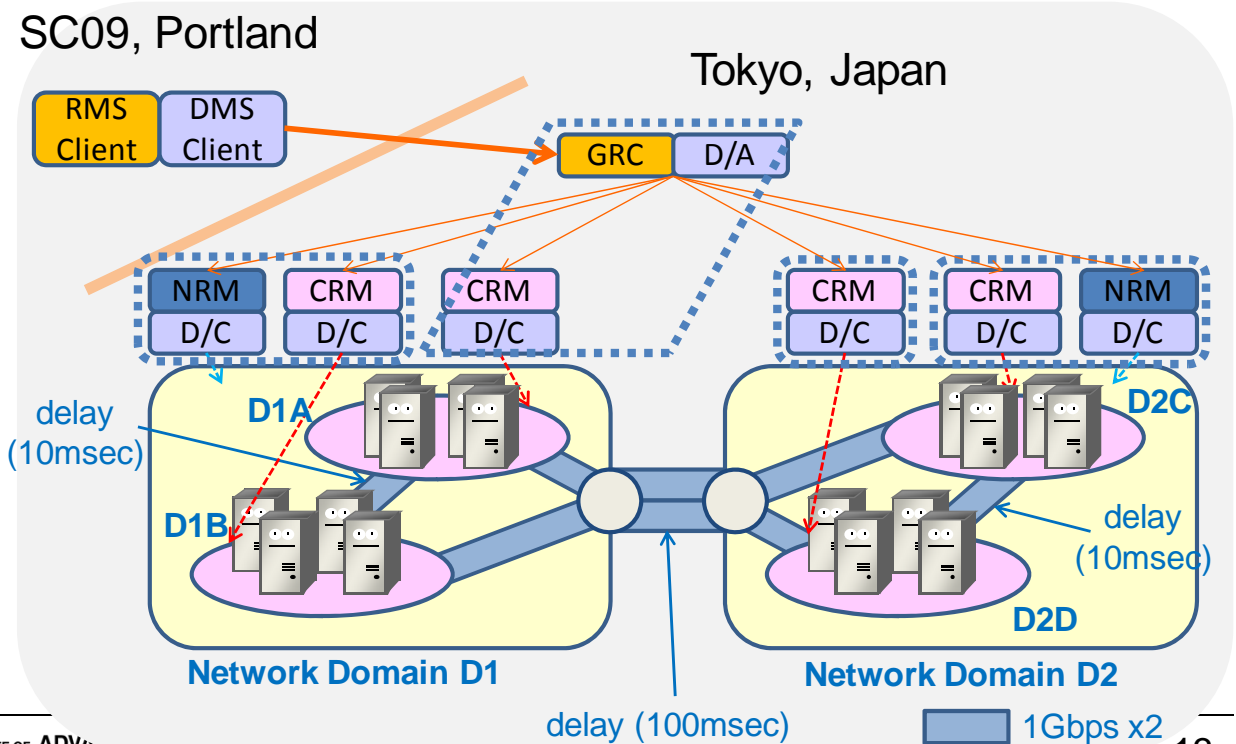
GridARSのエミュレーション環境での実験

• 実験概要

- RMSの基本性能
 - GT4版 vs. PureWS版
 - PureWS版複数RM
- DMSの基本性能
 - PureWS版複数RM
- SC09での実証実験
 - PureWS版複数RM

• エミュレーション環境

- Intel Core2 Quad Q9550(2.83GHz), CentOS 5.3, 4GBメモリ x 20台
- ネットワークエミュレータ GNET-1

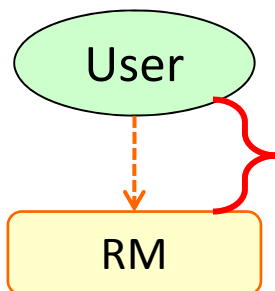


RMSの基本性能: GT4版 vs. PureWS版

オペレーション処理時間 [msec]

オペレーション	PureWS版	GT4版	GT4/GSI版
create	14.9	15.4	132.2
reserve	24.0	40.5	165.6
getCommandStatus	13.6	13.8	145.9
commit	13.8	15.2	157.1

GT4/GSI版は認証, SSLあり



PureWS版 vs. GT4版

- reserveはGT4版がやや長い, ほぼ同程度
- いずれも40msec程度以下で非常に高速

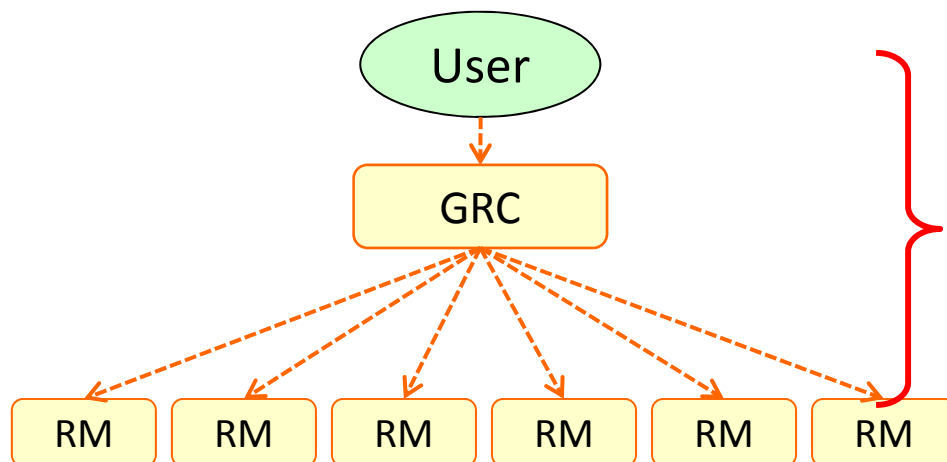
GT4版 vs. GT4/GSI版

- 認証, SSL のオーバーヘッドは150msec以下

RMSの基本性能: PureWS版複数RM

オペレーション処理時間 [msec]

オペレーション	RM x 1	RM x 2	RM x 3	RM x 4	RM x 5	RM x 6
create	14.7	13.3	12.7	12.4	12.3	13.4
reserve	22.9	21.2	19.8	20.1	20.2	21.9
getCS	12.2	11.0	10.1	10.0	10.3	10.7
commit	11.9	10.2	9.6	10.1	11.2	9.8

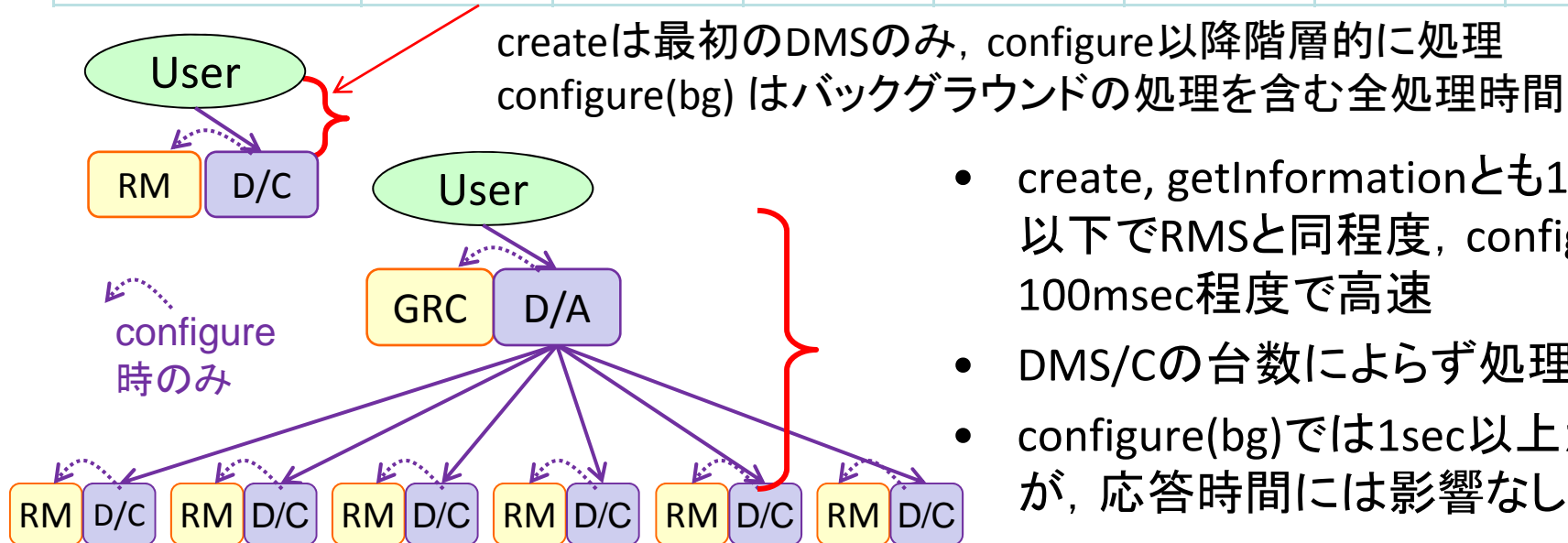


- RMの台数によらず、各オペレーションの処理時間は同程度
- GRCで複数RMへの要求をマルチスレッドで同時処理

PureWS版DMSの基本性能

オペレーション処理時間 [msec]

オペレーション	C-DMS/C	DMS/Cx1	DMS/Cx2	DMS/Cx3	DMS/Cx4	DMS/Cx5	DMS/Cx6
create	11.4	14.3	13.4	13.8	13.4	13.9	13.1
configure	38.5	68.1	85.7	115.7	116.2	119.9	116.1
configure (bg)		1078.1	1095.3	1126.9	1128.2	1132.6	1125.5
getInformation	3.1	11.1	11.3	11.6	13.3	10.7	13.0



- create, getInformationとも15msec以下でRMSと同程度, configureも100msec程度で高速
- DMS/Cの台数によらず処理可能
- configure(bg)では1sec以上かかるが, 応答時間には影響なし

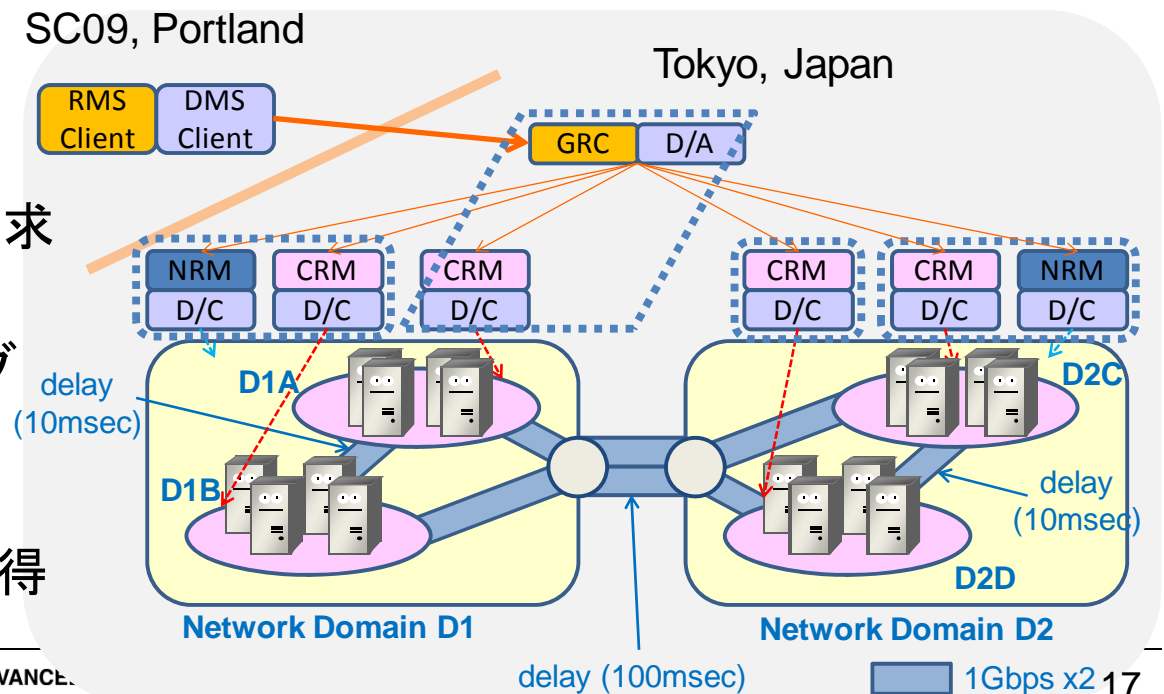
SC09での実証実験

- 2009年11月に米国ポートランドで開催されたSC09国際会議でGridARSの実証実験

→ PureWS版RMSとDMSの連携

実験内容

- RMSのCLIから資源要求
- GRCのスケジューラで SC09, Portland 資源選択・確保
- DMSのCLIから予約資源のモニタリング要求
- RMSとDMSが連携し、関連RMでモニタリングエージェントを起動
- 予約時刻にモニタリング情報を取得



関連研究

資源管理システム

- VIOLA
 - WS-Agreement標準を採用, GT4で開発
- ORCA (Open Resource Control Architecture), DRAC (Dynamic Resource Allocation Controller)
 - ネットワーク機器のマネージメントまでを対象

モニタリングシステム

- perfSONAR
 - いくつかのネットワークテストベッドでデプロイされている
 - ネットワーク資源専用のモニタリングツール
 - 情報を集中DBに格納

まとめと今後の課題

まとめ

- GNS-WSI3ベースのGridARS RMSとDMSを開発
 - GT4版: 高機能
 - PureWS版: インターオペラビリティ
- 開発したGridARS RMSとDMSを用いた実験
 - GT4版, PureWS版とも, RM数によらず高速に処理可能
 - SC09でRMSとDMSの連携を実証

今後の課題

- PureWS版の機能追加
- 実広域分散環境での実証実験の実施

謝辞

- 本研究の一部は、情報通信研究機構(NICT)の委託研究「ダイナミックネットワーク技術の研究開発」の助成を受けたものである