

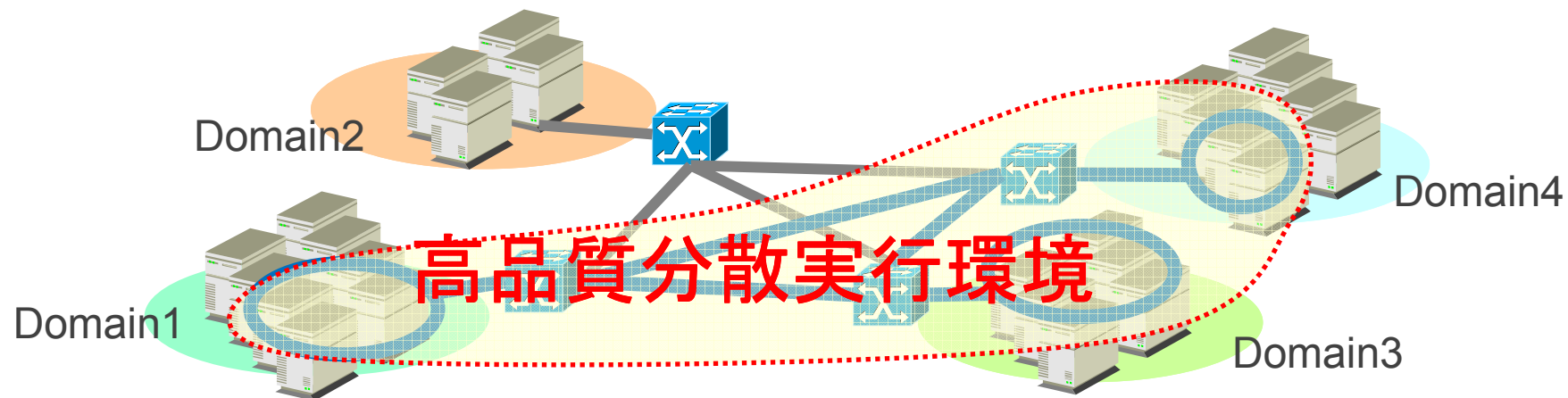
# 高品質分散実行環境のための 計算・ネットワーク資源の グローバルスケジューリング手法

竹房あつ子, 中田秀基,  
工藤知宏, 田中良夫

産業技術総合研究所

# 高品質分散実行環境と資源予約

- 帯域保証されたパケットネットワークと計算機，ストレージを統合した実行環境
- 複数資源の同時確保では資源予約が重要
  - KOALAグリッドスケジューラやQBETSバッチキュー予測サービスでは予約なしで同時確保→保証なし
  - 資源予約が確実な手段

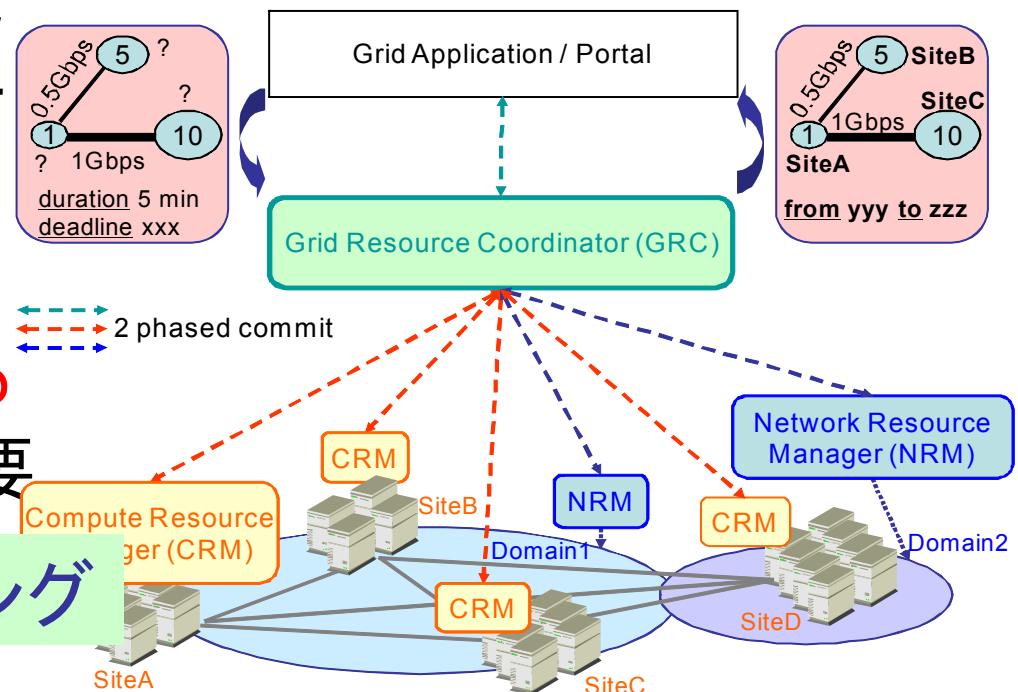


# GridARS資源管理フレームワーク

- 性能保証と事前予約機能を有する複数資源マネージャと連携し、高品質分散実行環境を構築
- 複数のグローバル資源コーディネータ(GRC)と資源マネージャ(RM)で構成

- ユーザの要求を満たす資源群を適切に確保しつつ、資源群の有効活用する**GRCにおけるスケジューリングが重要**

**グローバルスケジューリング**



# グローバルスケジューリングの課題

- 多様なスケジューリング指標
    - ユーザの観点
      - 早い時刻に確保
      - 価格の安い資源群を確保
      - 品質優先で確保
    - 資源管理の観点
      - 複数資源提供者に対して平等な負荷の分配
      - 特定の資源への優先割り当て(省エネ, 連携関係)
      - サービスレベルへの配慮
  - 資源群のトポロジの考慮も必要
    - 複数ドメインが存在
- 多様な指標に対応できる, 計算・ネットワーク資源を同時確保するグローバルスケジューリング手法

# 計算・ネットワーク資源のグローバル スケジューリング手法の提案

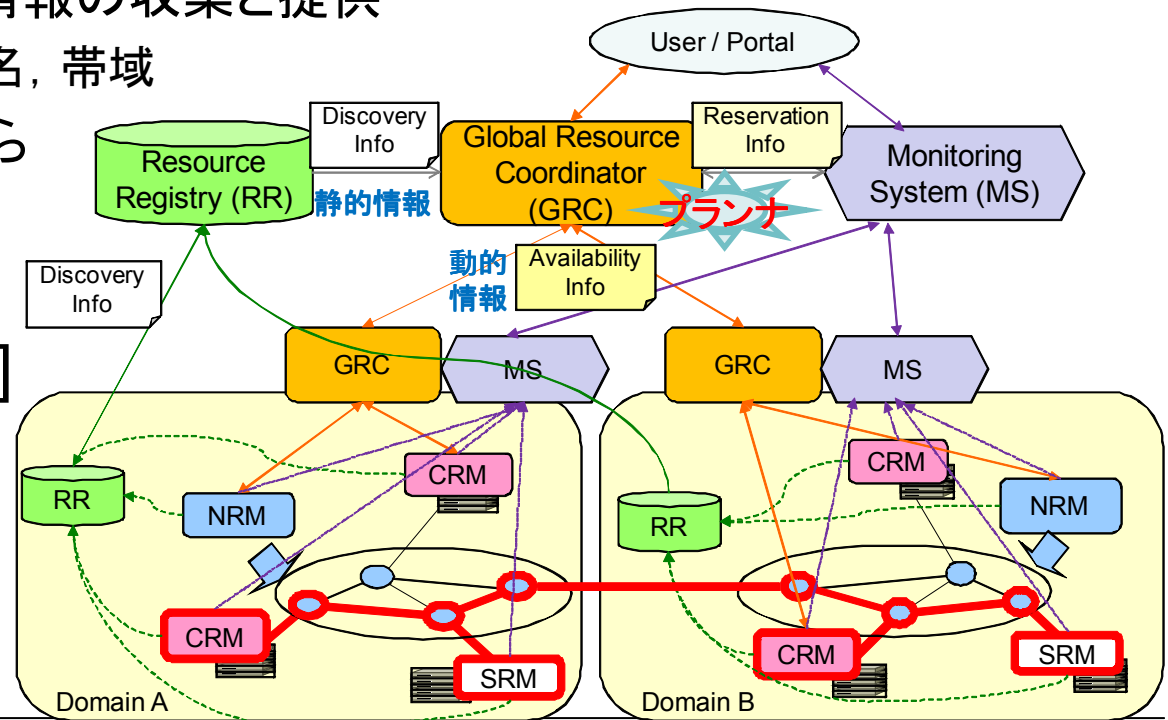
- 高品質分散実行環境の構築が目的
  - 計算・ネットワーク資源の同時確保
- オンラインスケジューリング手法
  - 動的資源情報をRM(資源マネージャ)から取得
  - 組合せ最適化問題に基づき、多様な指標に対応
  - 資源群とユーザの資源要求をグラフで表現
- シミュレーションによる評価
  - ユーザの要求と資源管理要件に対する有効性
  - 高い予約成功率

# 発表の概要

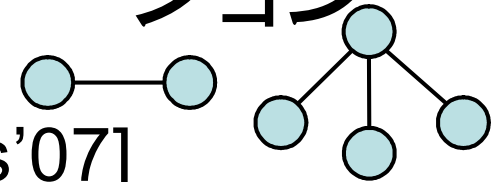
- グローバルスケジューリングモデル
  - 資源管理フレームワーク
  - ユーザの資源要求
- グローバルスケジューリング手法
  - スケジューリング手順と方針
  - 組合せ最適化に基づくスケジューリング
  - 事前予約への適用
- 提案手法のシミュレーションによる評価
- 関連研究
- まとめと今後の課題

# 資源管理フレームワーク

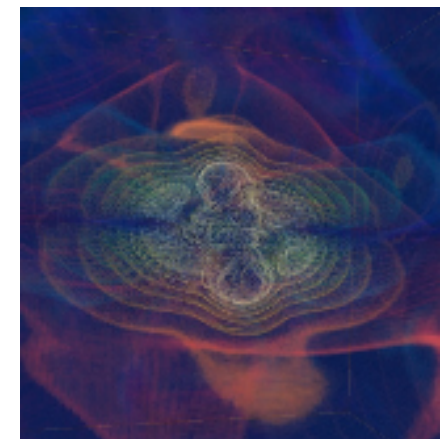
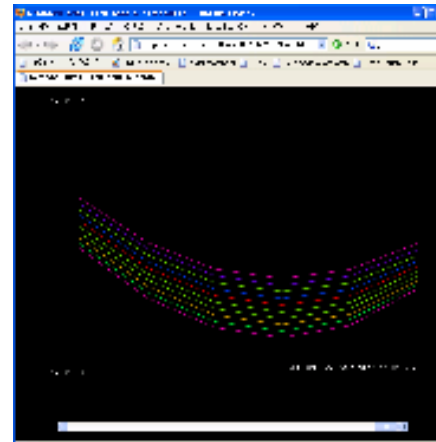
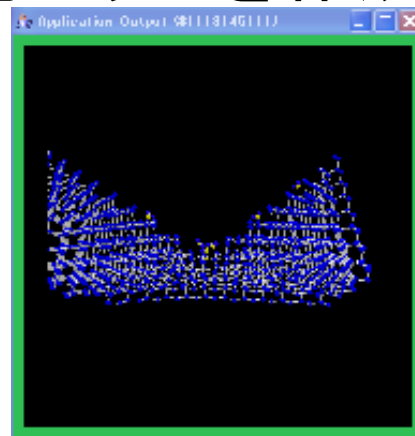
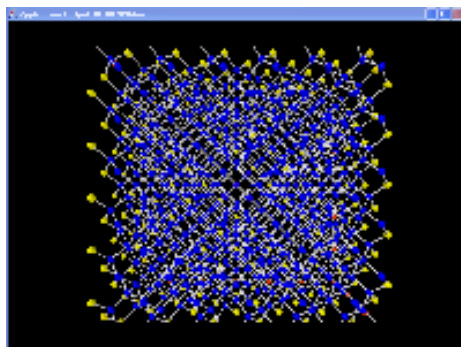
- 資源管理システム(RMS) [JSSPP'07]
  - GRCとRMが階層的に構成
  - **スケジューリング機能(プランナ)**を持つGRCがある
- 資源レジストリ(RR) [DIEM'09]
  - RMの所在, 静的資源情報の収集と提供
    - 総CPU/コア数, 拠点名, 帯域
  - 動的資源情報はRMから直接取得
- 資源モニタリングシステム(MS)[ComSys'09]
  - 確保した資源の利用状況を収集・提供
  - 再構成は手動



# 高品質分散実行環境アプリケーション



- アプリケーション事例[iGrid'05, GridNets'07]
  - GridRPCおよびGridMPIで実装された力学シミュレーション, 化学反応シミュレーション
  - ブラックホールの可視化
  - HDビデオデータライブストリーム
- 提案手法では**同時確保**のための予約プランを作成



# ユーザの資源要求

- 計算資源

- CPU/コア数(以降CPU数), 属性情報(OS)

- ネットワーク資源

- 帯域, 遅延, 属性情報

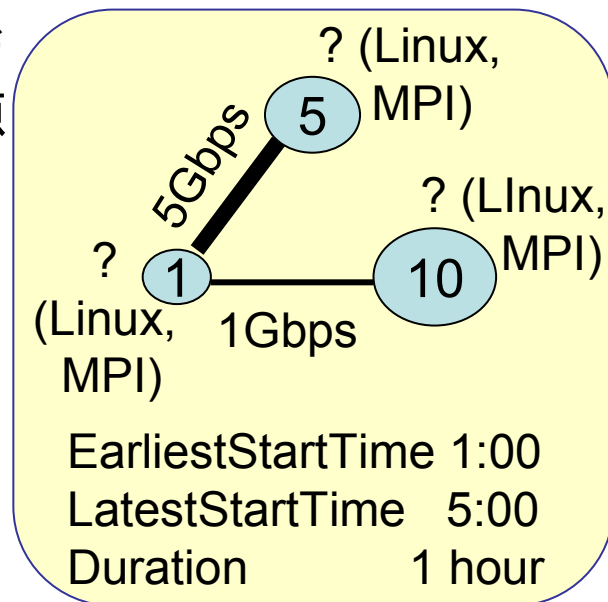
- 時刻

- 直接指定 / EST, LST, D

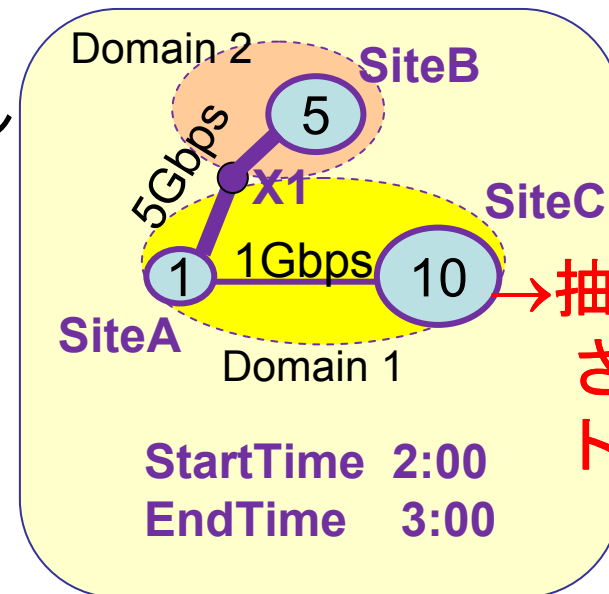
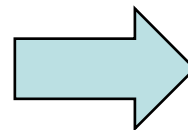
- EST: Earliest Start Time
- LST : Latest Start Time
- D : Duration

(締切時刻=LST + D)

ユーザの資源要求

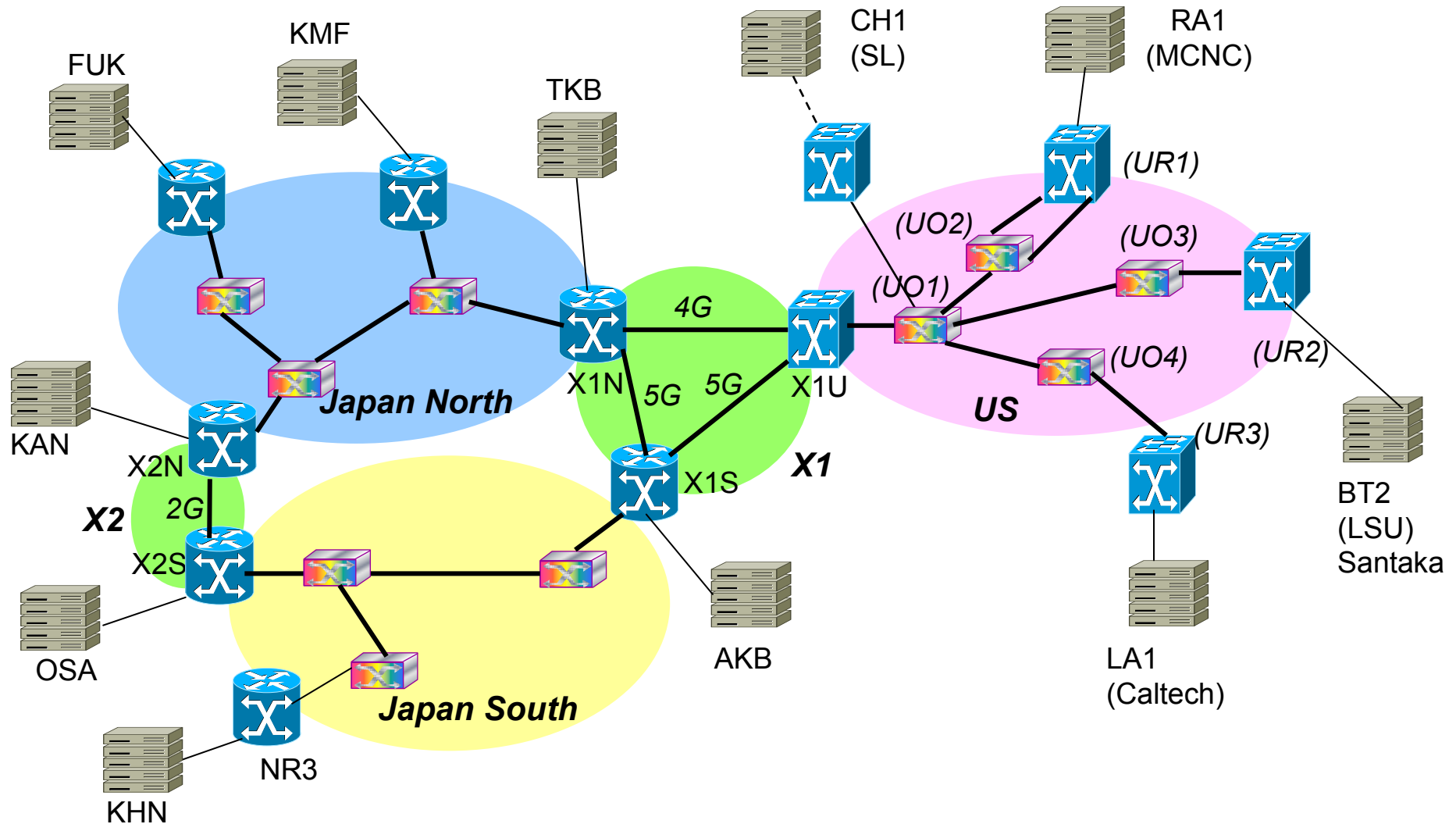


予約プラン



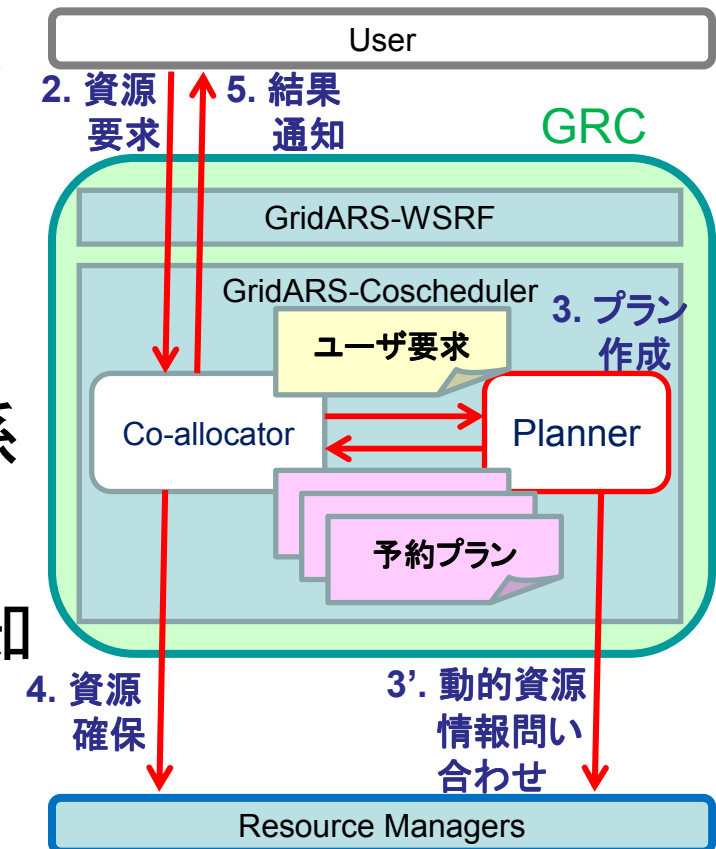
→ 抽象化されたトポロジ

# G-lambda, EnLIGHTened 実験環境



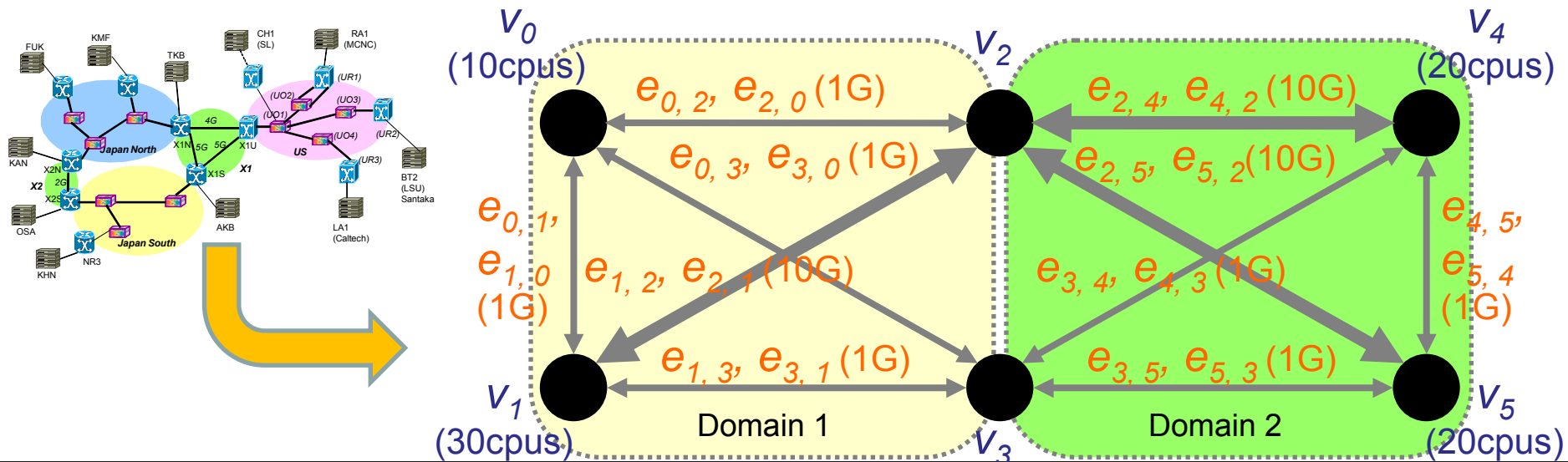
# グローバルスケジューリングの手順

1. 静的資源情報をあらかじめRRから取得
2. ユーザの資源要求を受け取る
3. **GRCのプランナが予約プランを複数作成**
  - プランナは関連する複数RMから動的資源情報を取得→ $O(1)$
4. 予約プランから, 複数RMと連係して資源確保
5. 確保成功 / 失敗をユーザに通知
  - 失敗の場合, ユーザは条件を変えて再度要求



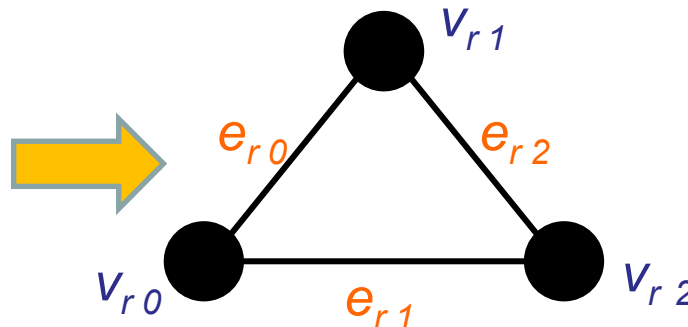
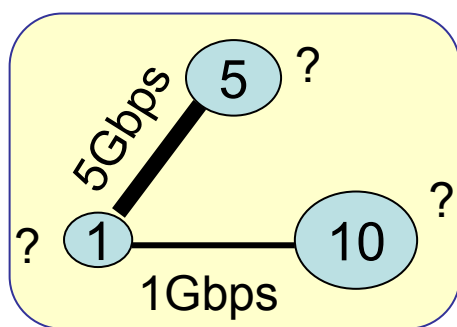
# 資源群のモデリング

- 資源群を有向グラフ  $G=(V, E)$  で表す
  - $V$ :  $G$ の点の集合,  $E$ :  $G$ の枝の集合
  - $v_q$ : 計算資源サイトまたはネットワークドメイン交換点
  - $e_{o,p}$ : 資源または交換点間のパス ( $o$ : 始点,  $p$ : 終点)
- パラメータ (枝のパラメータは双方向で共有)
  - $wc_i$  ( $i \in V$ ): CPU数,  $wb_k$  ( $k \in E$ ): 帯域
  - $vc_i$  ( $i \in V$ ): CPU数の価値,  $vb_k$  ( $k \in E$ ): 帯域の価値



# ユーザの資源要求のモデリング

- 資源要求を完全グラフ  $G_r = (V_r, E_r)$  で表す
  - $V_r$ : 要求資源サイトの集合
  - $E_r$ :  $V_r$ 間を結ぶ枝の集合
- 要求パラメータ
  - $rc_j$  ( $j \in V_r$ ):  $V_r$ に必要なCPU数
  - $rb_l$  ( $l \in E_r$ ):  $E_r$ に必要な帯域

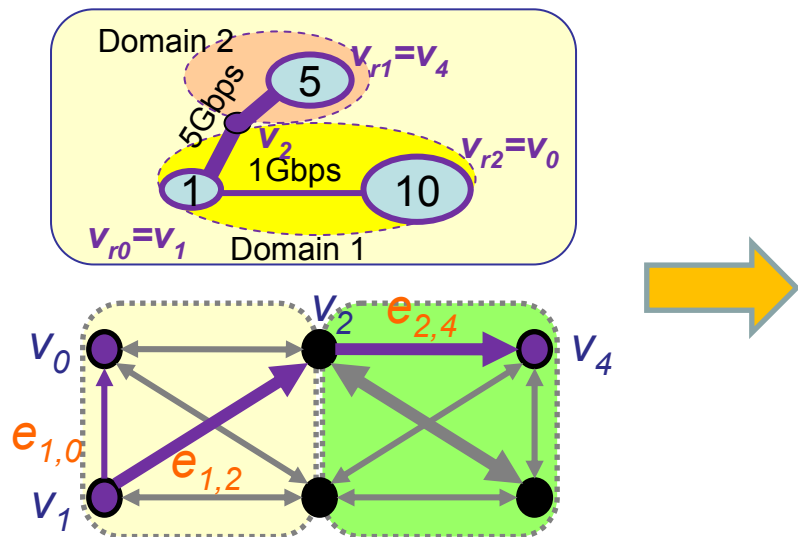


$$(rc_0, rc_1, rc_2) = \begin{matrix} V_{r0} & V_{r1} & V_{r2} \\ (1, & 5, & 10) \end{matrix}$$

$$(rb_0, rb_1, rb_2) = \begin{matrix} e_{r0} & e_{r1} & e_{r2} \\ (5, & 1, & 0) \end{matrix}$$

# 求める予約プラン

- $X : x_{i,j} \in \{0, 1\} (i \in V, j \in V_r)$
- $Y : y_{k,l} \in \{0, 1\} (k=(m, n) \in E, m, n \in V, l=(o, p) \in E_r, o, p \in V_r)$ 
  - 選択計算資源サイトの要素  $x_{i,j}$  の値は **1**, その他は **0**
  - 選択ネットワークパスの要素  $y_{k,l}$  の値は **1**, その他は **0**
  - **整数計画法で解く**



$$X = \begin{matrix} & v_0 & v_1 & v_2 & v_3 & v_4 & v_5 \\ v_{r0} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 \end{bmatrix} \\ v_{r1} & \begin{bmatrix} 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \\ v_{r2} & \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \end{matrix}$$

$$Y = \begin{matrix} & e_{0,1} & e_{1,0} & e_{1,2} & e_{2,1} & e_{2,4} & e_{4,2} & \dots \\ e_{r0} & \begin{bmatrix} 0 & 0 & 1 & 0 & 1 & 0 & \dots \end{bmatrix} \\ e_{r1} & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \\ e_{r2} & \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots \end{bmatrix} \end{matrix}$$

# 目的関数と制約条件

- 目的関数

- Minimize

$$\sum_{i \in V, j \in V_r} v c_i \cdot r c_j \cdot x_j + \sum_{k \in E, l \in E_r} v e_k \cdot r b_l \cdot y_{k,l} \quad (3)$$

- 制約条件

- Subject to

$$\forall j \in V_r, \sum_{i \in V} x_{i,j} = 1 \quad (4)$$

$$\forall i \in V, \sum_{j \in V_r} x_{i,j} \leq 1 \quad (5)$$

$$\forall i \in V, \sum_{j \in V_r} r c_j \cdot x_{i,j} \leq w c_i \quad (6)$$

$$\forall l \in E_r, \sum_{k \in E_r} y_{k,l} \begin{cases} \geq 1 & (r b_l \neq 0) \\ = 0 & (r b_l = 0) \end{cases} \quad (7)$$

$$\forall k \in E, \sum_{l \in E_r} r b_l \cdot y_{k,l} \leq w b_k \quad (8)$$

$$\forall l = (o, p) \in E_r, \forall m \in V, \sum_{n \in V, m \neq n} y^{(n,m), (o,p)} - \sum_{n \in V, m \neq n} y^{(m,n), (o,p)} = \begin{cases} x_{m,o} - x_{m,p} & (r b_l > 0) \\ 0 & (r b_l = 0) \end{cases} \quad (9)$$

$x_{i,j}, y_{k,l}$ を関連づける制約

(3) 計算・ネットワーク資源の総額を最小化((b)価格優先)

(4),(5),(6)は計算資源

(4) 選択サイトは1つのみ

(5)  $v_j$ は1回以上選択されない

(6) 要求CPU数が提供可能

(7),(8),(9)はネットワーク資源

(7) 要求パスがある場合,  $y_{k,l}$ の総和が1以上, ない場合0

(8) 要求帯域が提供可能

(9) **流量保存則**から導く

# 流量保存則の応用

- グラフ上の2点間の経路上の各点において、流入する流量と流出する流量の差が供給量と等しくなる  
→流量を1とすると

始点の供給量 = 1, 終点の供給量 = -1

通過点の供給量 = 0

- 式(9)では,

$\forall l = (o, p) \in E_r, \forall m \in V, \rightarrow$  各パス  $l = (o, p)$  ( $o$ :始点,  $p$ :終点), 点 $m$ に対して

$$\sum_{n \in V, m \neq n} y^{(n, m), (o, p)} - \sum_{n \in V, m \neq n} y^{(m, n), (o, p)} = \begin{cases} x_{m, o} - x_{m, p} & (rb_l > 0) \\ 0 & (rb_l = 0) \end{cases}$$

流出量総和 流入量総和 0/1/-1(要求帯域がある場合)

( $\rightarrow m$ が始点ならば、 $x_{m, o} = 1$ ,  $m$ が終点ならば $x_{m, p} = 1$ となる)

提案手法は予約なしのグローバルスケジューリングでも適用可能

# 事前予約への適用

- 確保する時間帯がEST, LST, Dで指定される場合の探索方法
  - G-lambdaプロジェクトのGNS-WSIIに基づく
    - 商用サービスが前提で, 予約表は公開されない
    - 時刻指定での問い合わせ
- **3. GRCのプランナが予約プランを複数作成**
  - 3a. [EST, LST+D]から**予約時間帯候補を等間隔にN個選ぶ**
  - 3b. 関連するRMIに対し, N個の時間帯の動的情報を取得
  - 3c. 3bの情報から, 組合せ最適化の手法を適用し,  **$n(\leq N)$ 個の予約プランを作成 (独立して計算できるため $O(1)$ )**
  - 3d. 決定した **$n$ 個の予約プランの優先順位を決定**

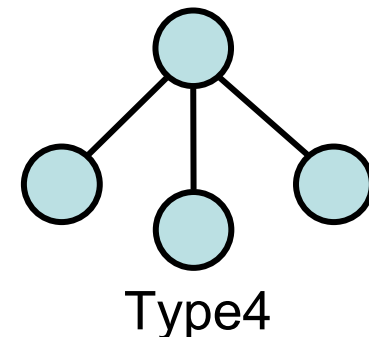
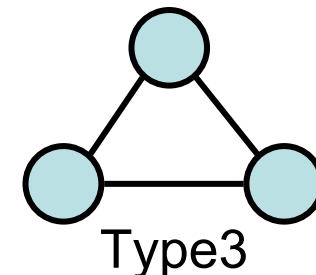
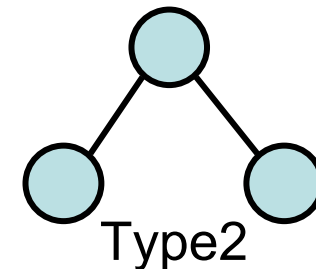
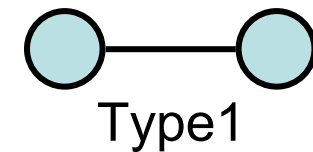
# グローバルスケジューリング方針の適用

- ユーザ
  - (a) 時刻優先 → 手順3dで順位付け
  - (b) 価格優先 → 式(3)の目的関数と手順3d
  - (c) 品質優先 → 式(3)の目的関数を変更と手順3d
- 資源管理者
  - (A) 複数RMへの平等な負荷分配 → 資源群のモデリングで重みを追加と目的関数を追加
  - (B) 特定資源の優先 → 資源群のモデリングで重みを追加と目的関数を追加
  - (C) ユーザのサービスレベルの考慮 → 手順3bで動的資源情報のフィルタ



# シミュレーションモデル (1/2)

環境設定	
RMS構成	GRM=1, NRM=3, CRM=10
サイト数/ドメイン	4/JN, 3/JS, 3/US
ドメイン交換点	X1(JN,JS,US), X2(JN, JS)
CPU数/単価	JN{8, 16, 32, 64}, JS{8, 16, 32}, US{8, 16, 32}/1
帯域 [Gbps] / 単価	ドメイン内=10/5, 交換点接続=20/3
資源要求設定	
ユーザ	UserA, UserB
資源要求の種類	<b>Type 1, 2, 3, 4</b> (一様分布)
平均要求到着間隔 [min]	5, 10 (ポアソン到着)
予約時間幅 [min] (D)	30, 60, 120
LST - EST	<b>予約時間幅</b> ×3
予約CPU数	1, 2, 4, 8 (一様分布)
予約帯域 [Gbps]	1

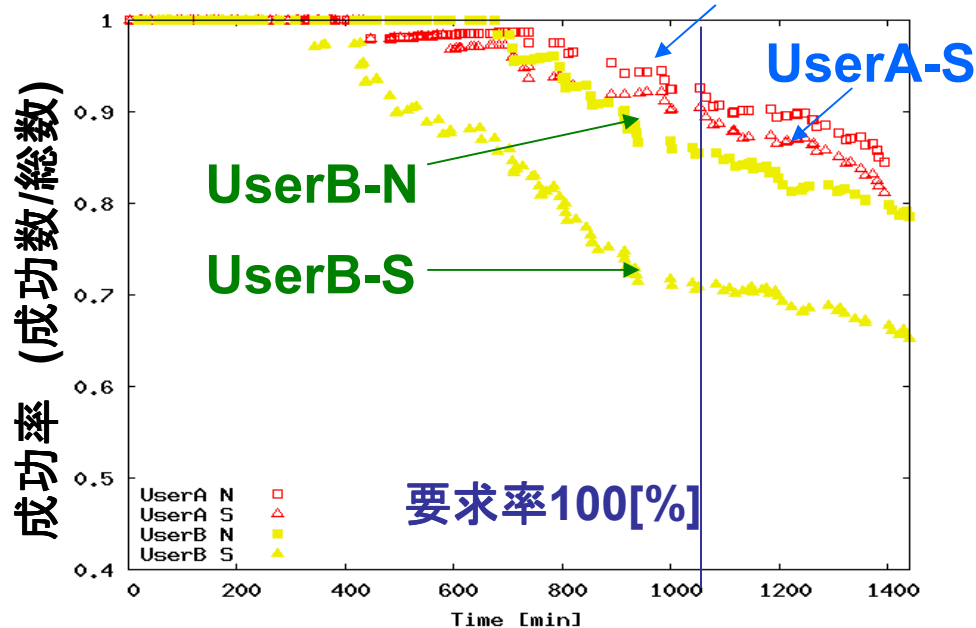


# シミュレーションモデル (2/2)

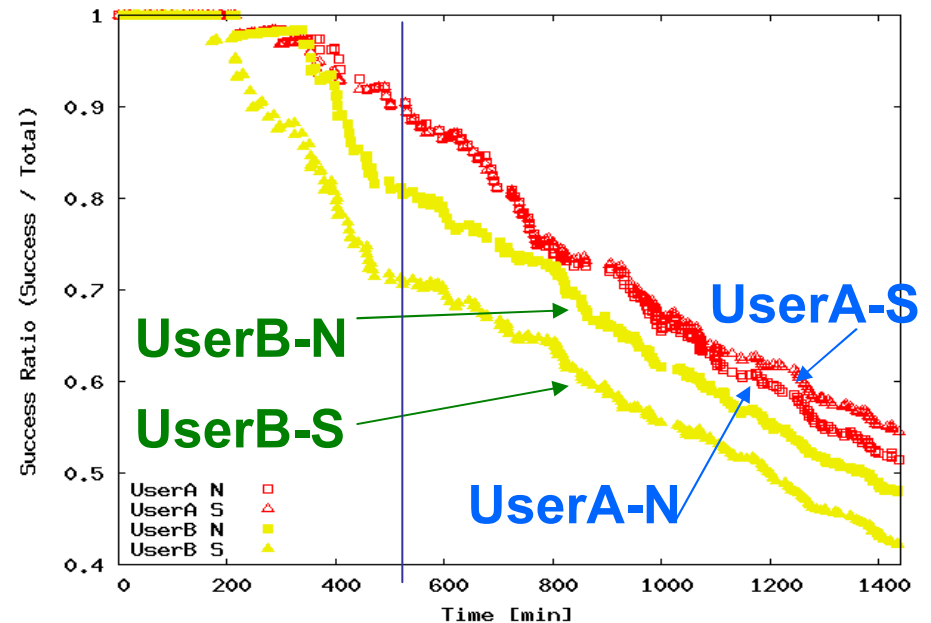
- 最初の24時間までに各ユーザの資源要求が到着し、それぞれ次の24時間の時間帯の資源予約をする (一様分布)
- GRCでの予約時間帯候補数  $N=10$
- (a)時刻優先で、目的関数は(b)価格が最小となるものとする
- サービスレベル(SL)による影響も調査
  - UserBはサービスレベルが低い場合も想定
- 組み合わせ最適化問題のソルバはGLPK (GNU Linear Programming Kit)を利用

# 評価結果

平均到着間隔=10 [min] **UserA-N**



平均到着間隔=5 [min]



- SLなし(-N)の場合は, UserAとUserBの成功率は同程度
- SLあり(-S)では, 10[min]のときで0.79→0.65まで低下  
→ユーザごとのSLを制御可能
- 資源要求率100[%]のとき, SLなしでUserA, Bとも0.85以上の成功率  
→提案手法は高い予約成功率

# 関連研究

- VIOLA[PPAM2005, Wäldrichら]
  - 計算・ネットワーク資源の事前予約ベースシステムを提案
  - スケジューリング方針はジョブのmakespanを小さくすることのみ
- Co-reservationのためのグローバル最適化[CoreGrid'08, Röblitz]
  - 最適化問題に基づく複数資源の同時予約手法を提案
  - ネットワークのモデル化が単純
- バックトラック法によるグローバルスケジューリング [SIGHPC'07, 安藤ら]
  - 計算・ネットワーク資源確保のためのアルゴリズムを提案
  - バックトラック法による資源探索のため, 適切な解の探索精度は低い
  - ワークフローにも対応

# まとめと今後の課題

- 計算・ネットワーク資源のグローバルスケジューリング手法を提案
  - 高品質分散実行環境の構築が目的とした同時確保
  - オンラインスケジューリング手法
    - 動的資源情報をRM(資源マネージャ)から取得
    - **組合せ最適化問題**に基づき、多様な指標に対応
  - シミュレーションによる評価
    - ユーザの要求と資源管理要件に対する有効性
    - **高い予約成功率**
- 今後は多様な環境, 指標を想定した詳細な評価

# 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」により実施した