



Grid Datafarmにおけるスケジュー リング・複製手法の性能評価

竹房あつ子 (お茶の水女子大学)

建部修見 (産業技術総合研究所)

松岡聡 (東京工業大学/国立情報学研究所)

森田洋平 (高エネルギー加速器研究機構)

データグリッド

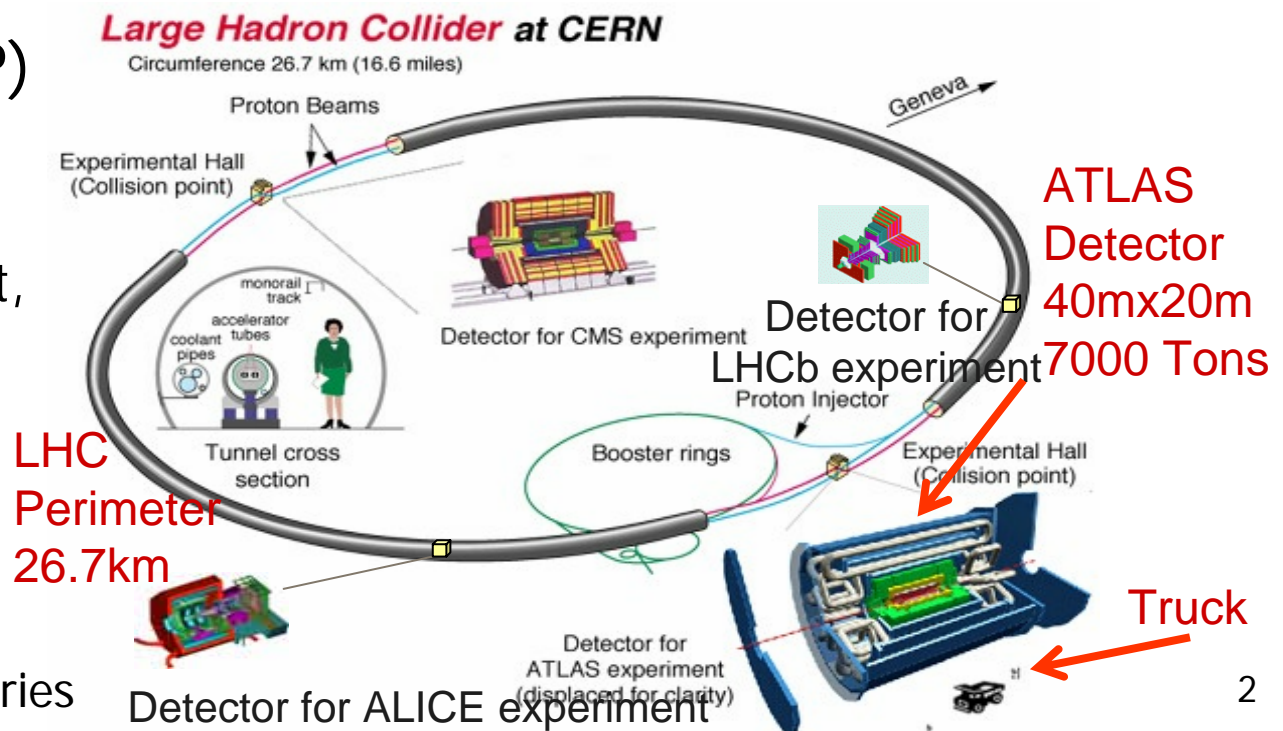
- グリッド技術を基盤にした大容量データに対する偏在するアクセスを可能にする技術
- 例: CERN Large Hadron Collider(LHC)実験 (2007年)

- 高速ファイル転送(GridFTP)

- データ複製管理(Globus Replica Mngmnt, Giggie)

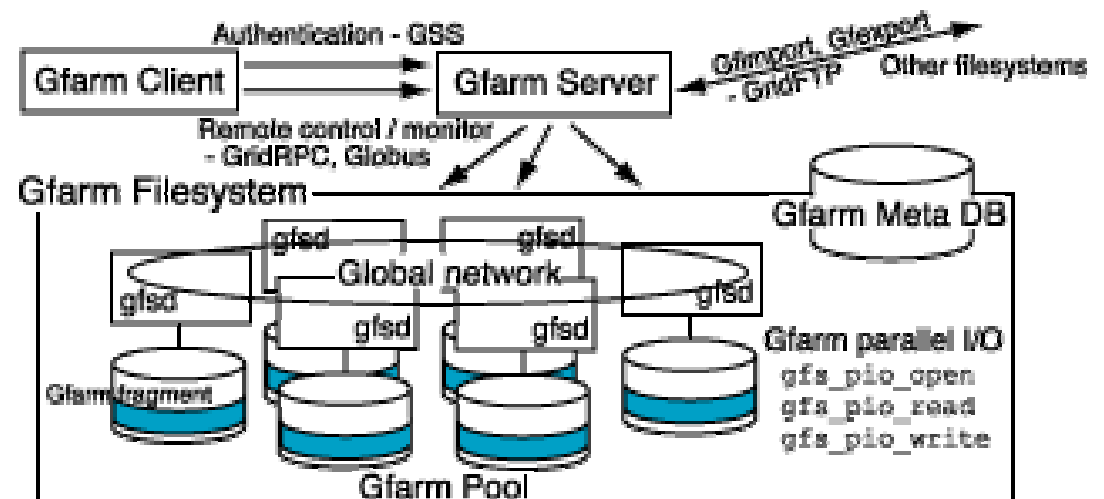
- 大規模データ高速アクセス

~2000 physicists
from 35 countries

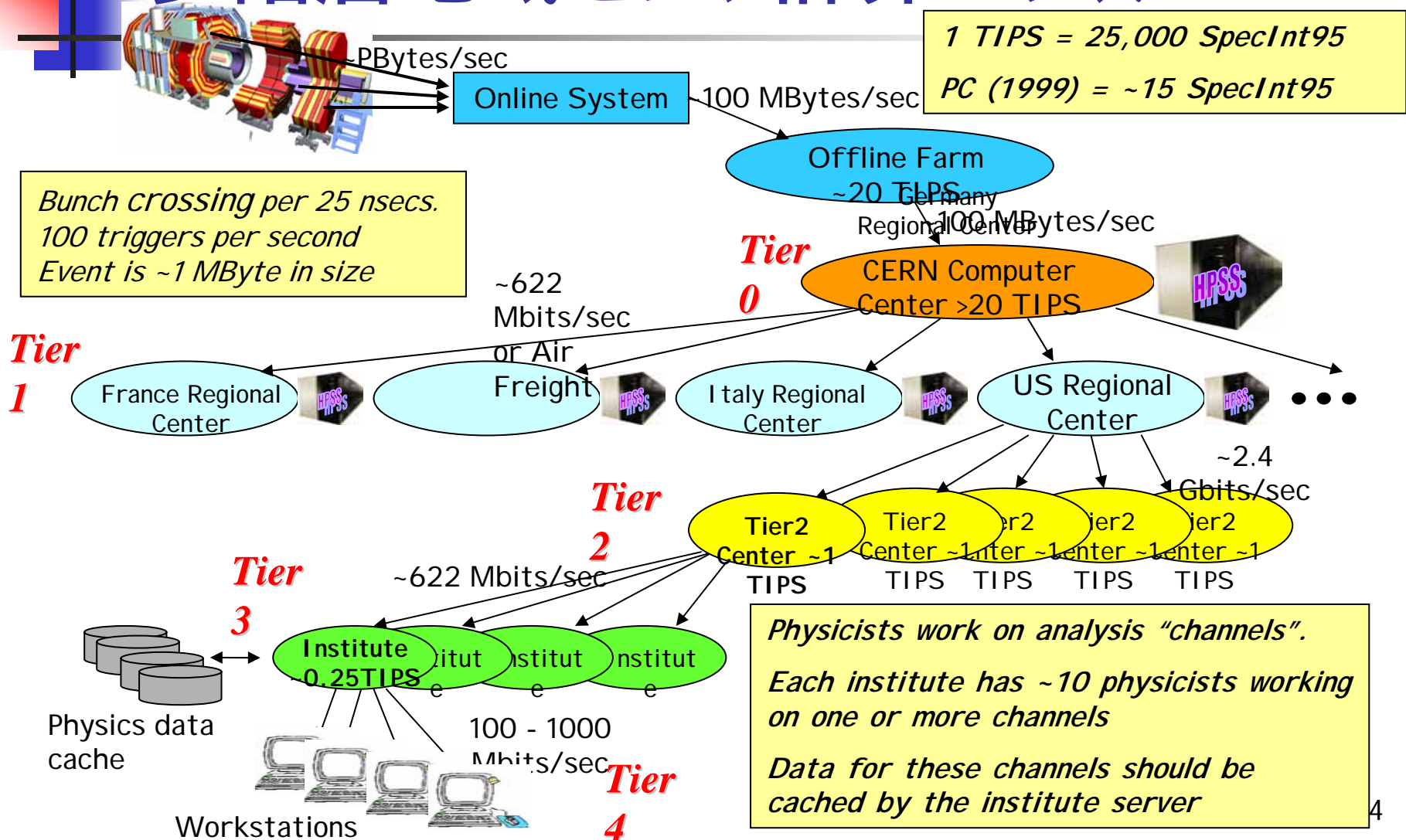


高速データアクセス

- (既存技術) HPSS, クラスタファイルシステム:
 - 計算ノードとI/Oノードが分離
 - スケーラブルなローカルI/Oバンド幅の実現が困難
 - 数GB/secのバンド幅で数PBのデータの読み込みに約12日間かかる!
- Grid Datafarmアーキテクチャ[産総研, 高エネ研, 東工大, 東大]:
 - 計算ノードとI/Oノードを融合
 - 局所参照性のあるデータアクセスに対し > TB/sec



LHCのためのMONARC型 多階層地域センタ計算モデル

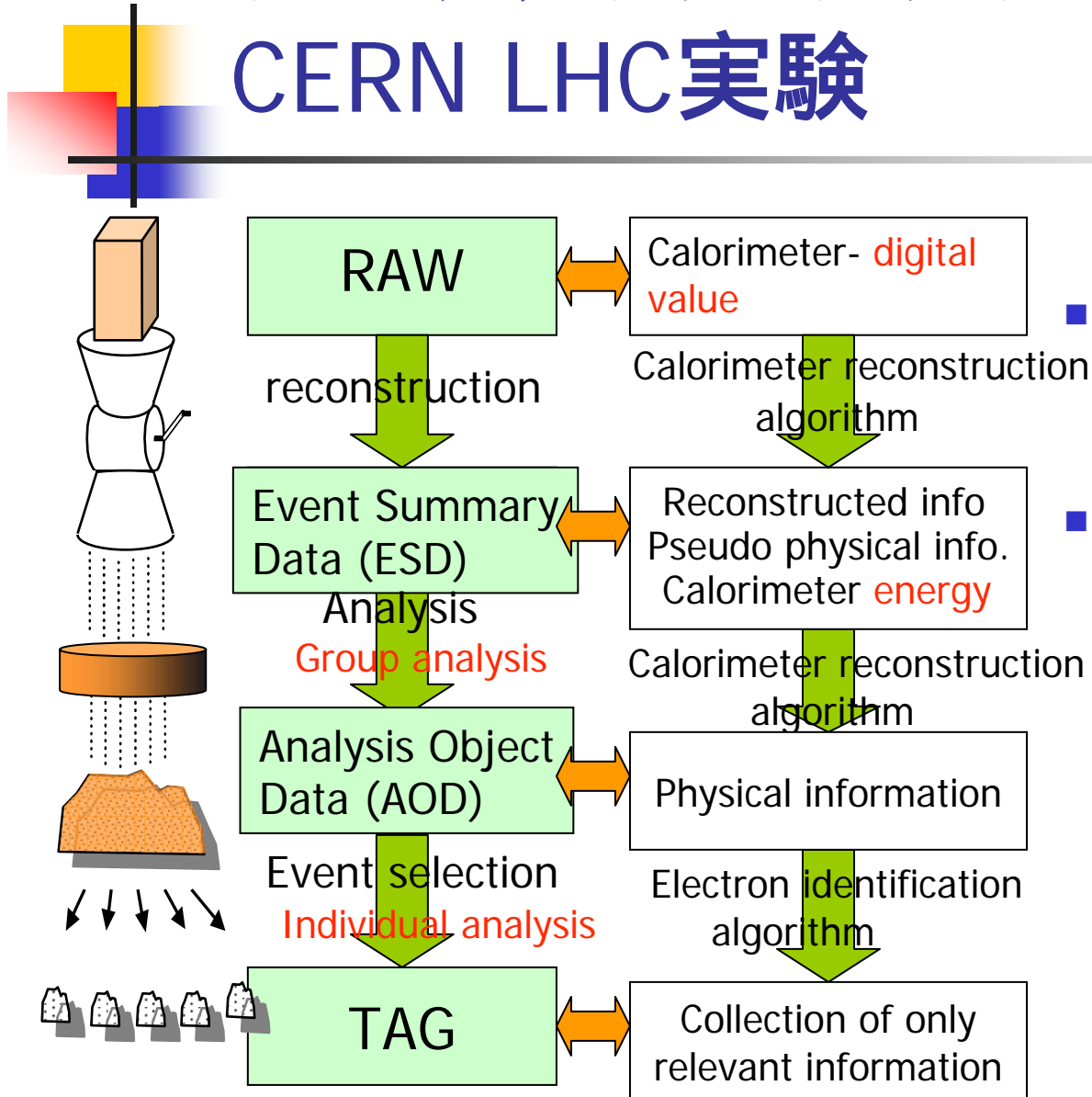




Grid Datafarmにおけるスケジューリング・複製手法の性能評価

- 2007年のLHC実験のパラメータを利用
- Bricksグリッドシミュレータによる評価
- Grid Datafarmアーキテクチャを想定
- データグリッドモデルの比較
 - MONARC型複数サイト分散クラスタ
vs. 単一サイト巨大クラスタ
- スケジューリングとデータ複製手法の比較
 - Owner Computes + バックグラウンドデータ複製
vs. MCT + オンデマンドデータ複製

データグリッドアプリケーション: CERN LHC実験



- ATLAS検出器から粒子衝突の観測データを得て, 解析

- 段階的な解析

- RAW→ESD (**Large**)
 - 2-4 times/year
- ESD→AOD (**Medium**)
 - once/month
- AOD→TAG (**Small**)
 - once/4 hours

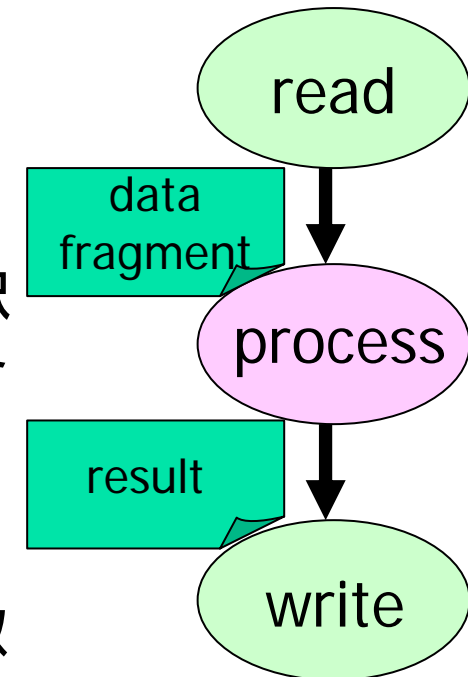
シミュレーションモデル: ジョブ処理

- LHC実験ジョブの処理手順:

- ユーザ(物理学者)はクライアント計算機から解析ジョブを投入
- データグリッドスケジューラは適切なサーバ群を選択
- ローカルにジョブが必要とするデータがなければ, 各サーバはデータ断片をダウンロード
- サーバは割り当てられたタスクを処理
- サーバは出力データを指定されたディスクに送信
- (クライアントは計算で得られた統計情報のみ受け取る→非常に小さいので無視できる)

- ジョブ処理に要する時間:

$$T_{response} = T_{read} + T_{process} + T_{write}$$



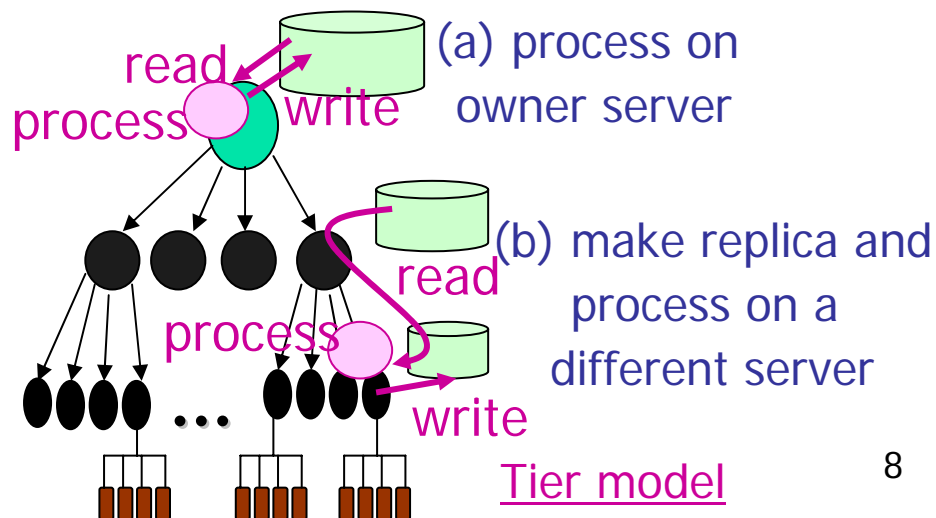
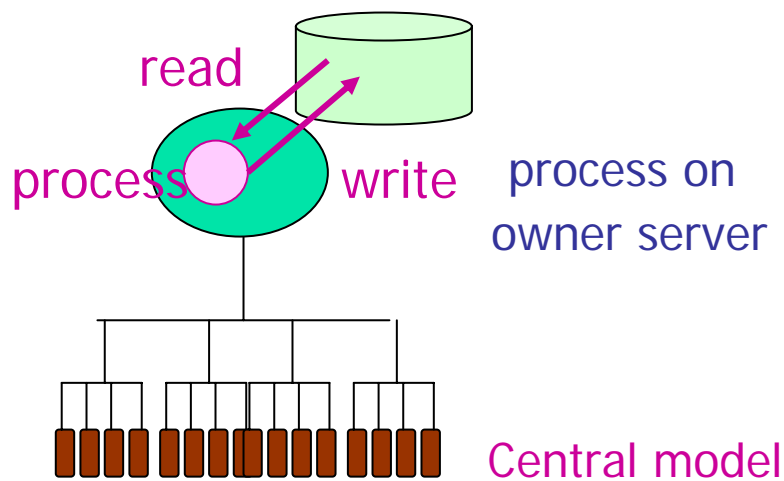
データグリッドモデル

■ Centralモデル:

- 単一巨大サイトで全てのジョブを処理
- 性能面, 管理面で効率よいが設備・管理コスト大

■ Tierモデル (MONARC型):

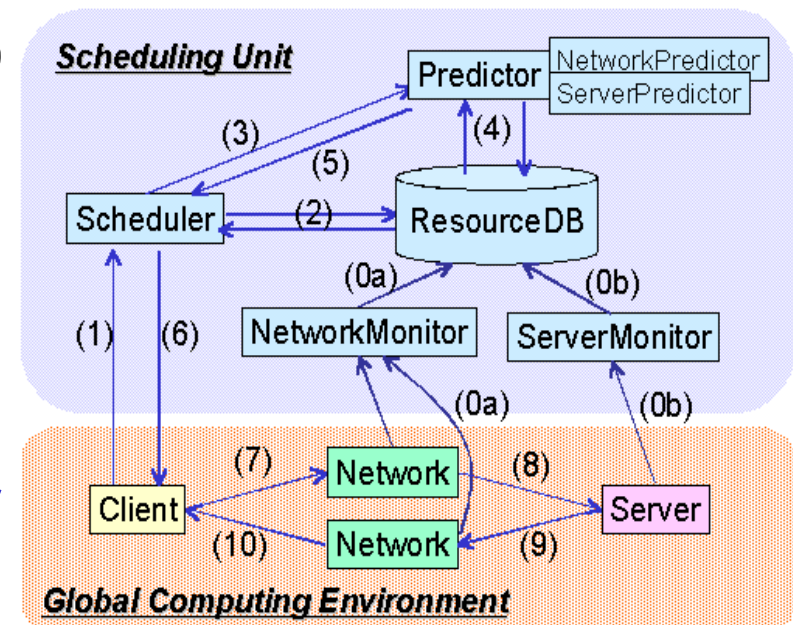
- 階層的な地域センタでジョブを分散処理
 - スケジューリングとデータ複製手法を利用
- 1拠点での電力, 予算, 管理コストの削減



Bricksグリッドシミュレータ

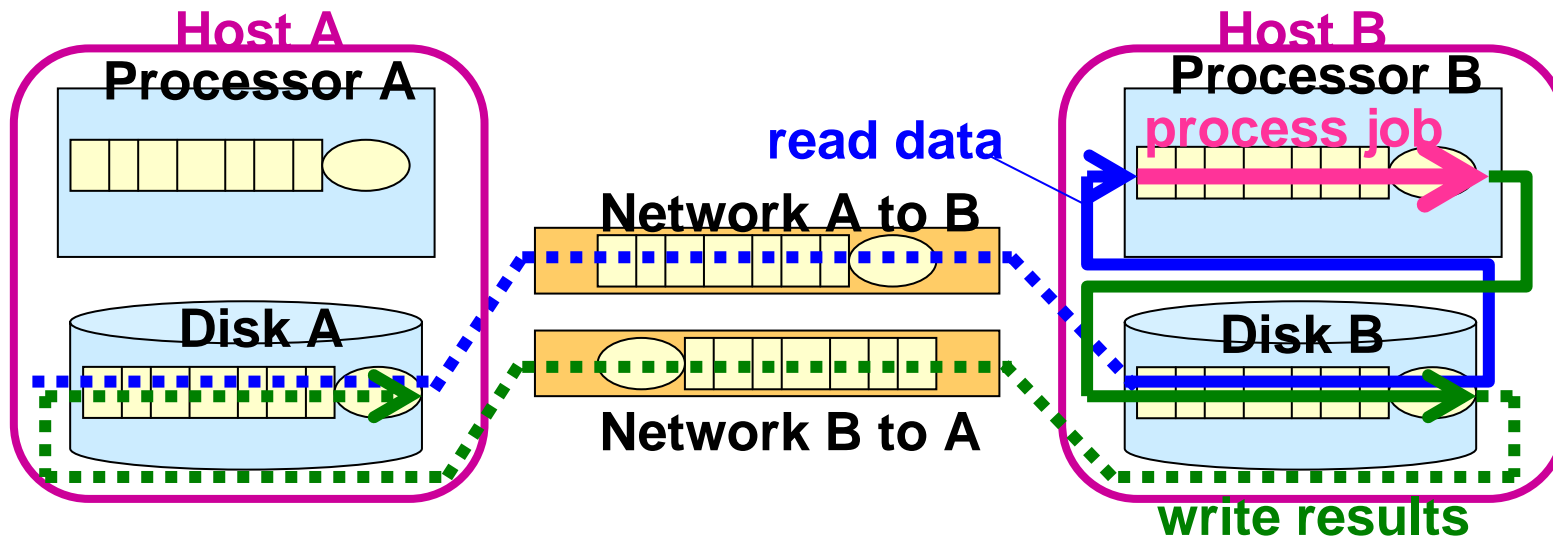
- Javaベースの離散イベントシミュレータ
- ネットワーク, サーバを待ち行列で表現
- 一般的なグリッドスケジューリングモジュールの提供
- 柔軟なシミュレーション環境設定
 - グリッドトポロジ
(e.g. 階層ネットワークトポロジ)
 - サーバ, ネットワークモデル
 - クライアントモデル
- 動的なグリッド環境で様々なスケジューリング手法の性能解析が可能

<http://grid-team.is.titech.ac.jp/bricks/>



Bricksのデータグリッド拡張

- **複製マネージャ**のスケジューリングユニットへの追加
- ファイル管理機構
- ディスクの表現 – 待ち行列を利用
 - 実線: サーバにジョブに必要なデータがある場合
 - 実線+破線: 計算サーバとジョブが格納されていない場合





Tierモデルのための スケジューリングとデータ複製手法

- Centralモデルではファイルアフィニティスケジューリング(ディスクowner-computerルール)を適用
- Tierモデルのためのスケジューリングとデータ複製手法
 - オンラインスケジューリング手法 (→オンデマンド複製)
 - データ複製手法 (→バックグラウンド複製)



オンラインスケジューリング手法

スケジューラはDataSourceHost(I/Oホスト),
ComputeHost(計算ホスト),
DataDestinationHost(I/Oホスト)を決定

- Greedy(MCT: Minimum Completion Time)
ジョブの応答時間が最短となるホストを選択
- OwnerComputes
ジョブ処理に要するデータを格納しているホストからMCT
となるホストを選択
- LoadBound-Read/-Write
指定した性能を超えるホストから, MCTとなるホストを選択
(I/Oホストと計算ホストが異なる場合は, -Readでは読み
込み時に, -Writeでは書き出し時に適宜複製を作成)



バックグラウンド複製手法

- 複製マネージャは定期的にシステム上の計算ホストの状況を調べ、適宜複製を作成する
- **LoadBound-Replication:**
 - データを格納しているホストに対し *Perfestimated* を算出
$$Perfestimated = Perf / (LoadAvg + 1)$$
 - $Perfspecified > Perfestimated$ の場合、*Perfestimated* が最小のホストから最大のホストへアクセス率 *AR* の高いデータの複製を生成する
$$AR = Naccesses / (Tcurrent - Tstored)$$



複製削除アルゴリズム

複製の作成でデータグリッドシステム上のディスク領域が不足した($x\%$ のホストの空きディスク領域が $y\%$ になった)場合, 複製を削除する

1. システム上に複製を持つデータのリストを作成
2. 1のリストを最後にアクセスされた時刻(LRU)が古い順にソート
3. 2のリストの最初からN個のデータに対し, アクセス率 $ARelim$ を算出 ($Ncopies$ は複製の総数)
$$ARelim = Naccesses / (Tcurrent - Tstored) / Ncopies$$
4. 以下の条件を満たすまで最初のN個のデータから, $ARelim$ が最小となるデータを削除 ($Compactness$ は削除頻度を決定するパラメータ)
$$TotalDiskSize \times Compactness > AvailableDiskSize$$
5. 4の条件が満たされなければ, 3に戻る
(本シミュレーションでは $Naccesses$ を10とした)

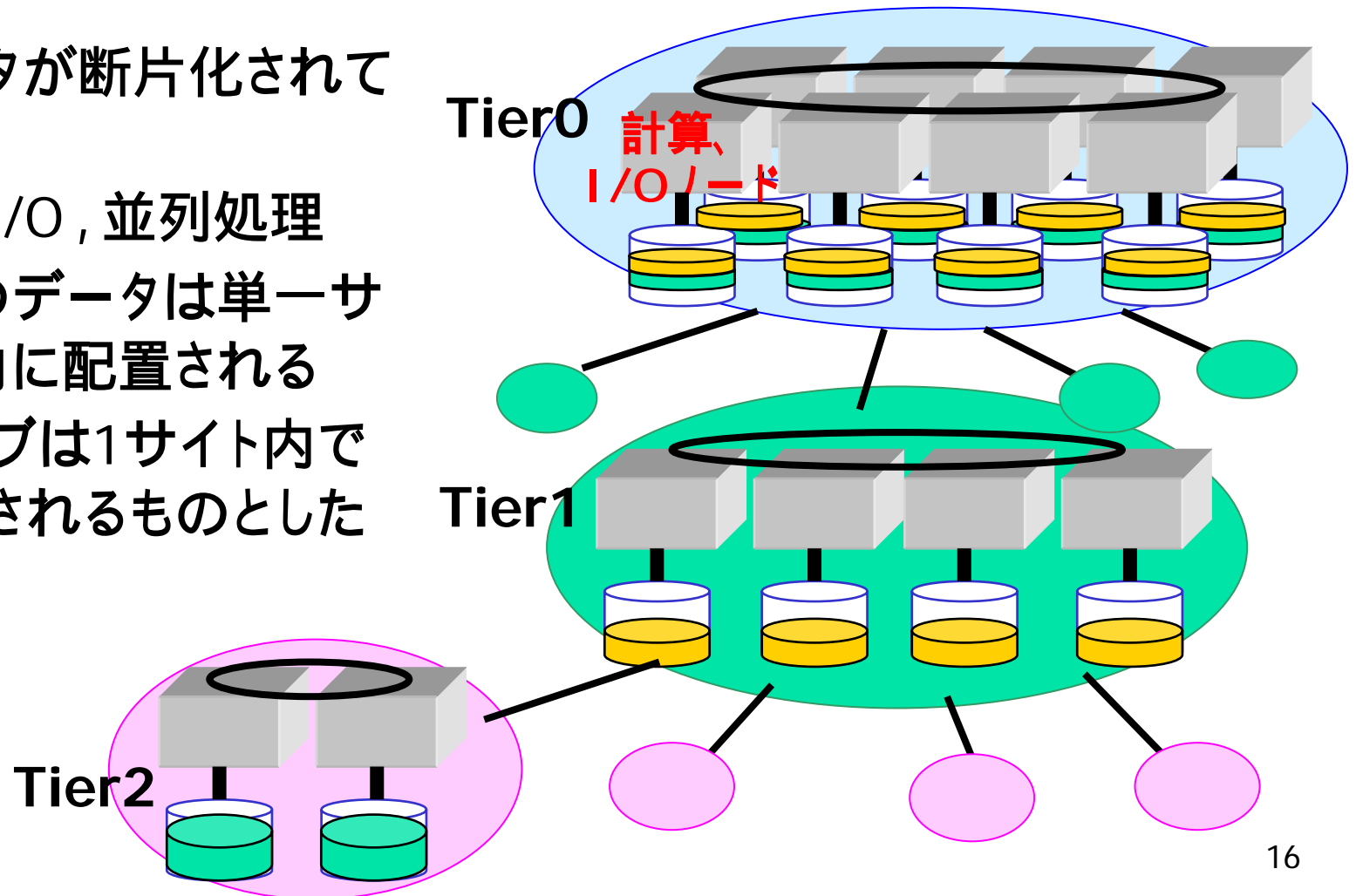


シミュレーションによる評価

- LHC実験を想定した性能評価
 - Grid Datafarmアーキテクチャ
 - Centralモデル vs. Tierモデル
 - Tierモデルでは, 5通りのスケジューリング・複製手法
 - データアクセスの局所性(ランダム/**時間的局所性**)
 - Bricksグリッドシミュレーションフレームワークを利用
- 評価指標
 - **平均応答時間**
 - ネットワークバンド幅消費量
 - 計算サーバ平均負荷値
 - データ断片の分散 (複製の数)

Grid Datafarmシステムの性能 評価設定

- データが断片化されて管理
- 並列I/O, 並列処理
- 1つのデータは単一サイト内に配置される
- 1ジョブは1サイト内で実行されるものとした



シミュレーション設定: 環境

モデル	ディスク容量 [PB]	サイトの総性能 [MSpecInt95]	サイト内 ノード数	総I/Oバンド幅
Central	2	0.5-1.8	10000	1[TB/sec]
Tier	T0 (x1): 2	0.6/0.5/ 0.4	10000	1[TB/sec]
	T1 (x4): 1	0.3/0.25/ 0.2	5000	500[GB/sec]
	T2 (x16): 0.1	0.03/0.025/ 0.02	500	50[GB/sec]

- Tierモデルの性能比はGriPhyNシミュレーションと同じ
- 待ち行列理論でのCentralの平均応答時間の見積もり
38.575-1.337 [hours] , 0.453318[MSI95]で飽和
- WAN/ローカルI/Oバンド幅は10[Gbps]と100[MB/sec]
- Grid Datafarmアーキテクチャでは,
総I/Oバンド幅=ローカルI/Oバンド幅×サイト内ノード数

シミュレーション設定: ジョブ

Job	# events / Job	計算サイズ [GSI95*sec]	頻度 (Avg.)	入力サイズ[TB]	出力サイズ [TB]
Large: RAW→ESD	1G	1000	1/4 [月]	1000	100
Medium: ESD→AOD		25	1/1 [月]	100	10
Small: AOD→TAG		5	1/4[時間]	10	0.1

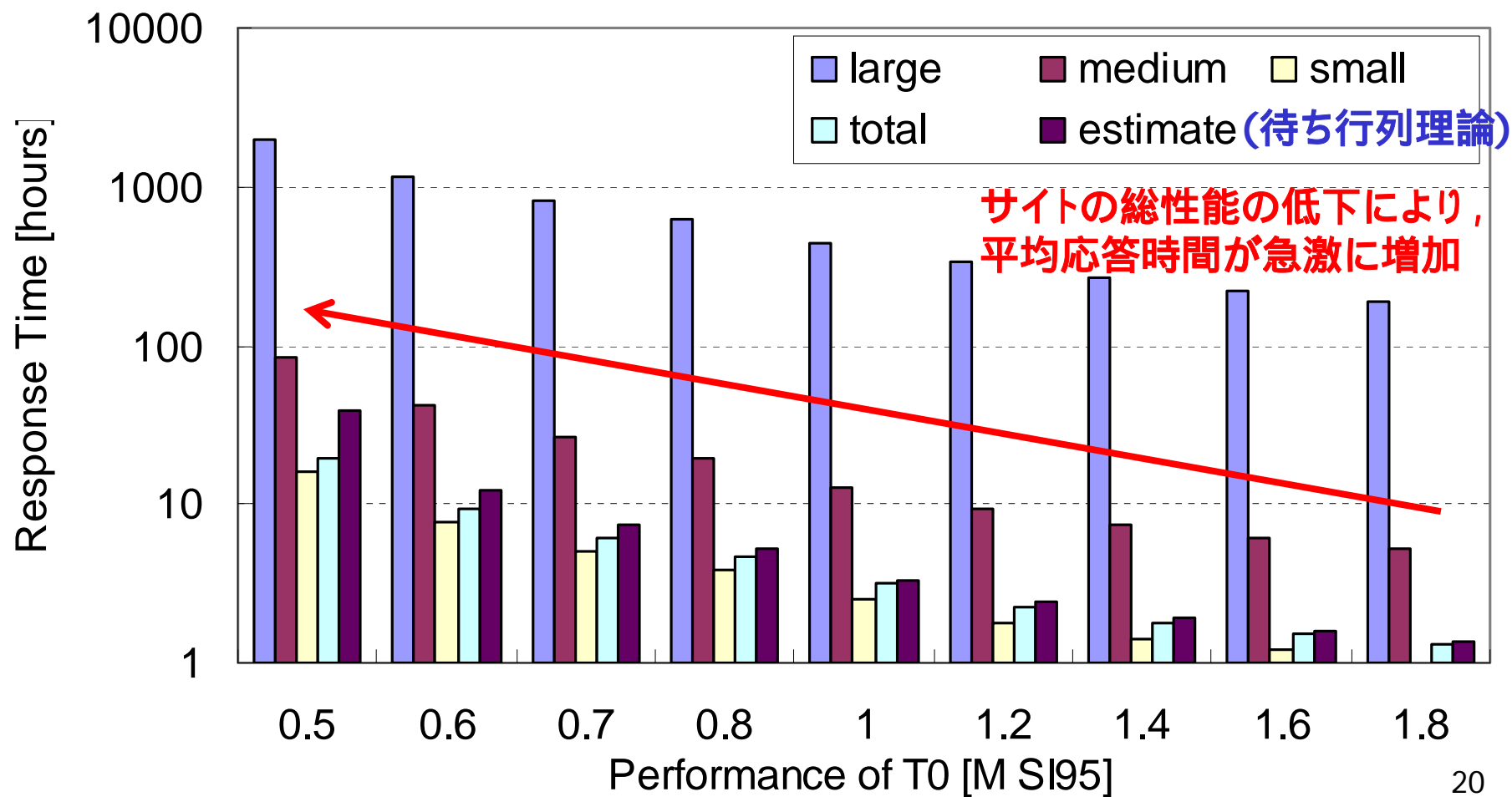
- MONARCレポートで挙げられたLHC実験での実パラメータ
- 全シミュレーションでTier0サイトにデータ (1PBx1, 100TBx2, 10TBx4) を格納
- 各アルゴリズムに対し, 1年間のシミュレーション×10を実施
- 東工大Presto IVクラスタ (Dual Athlon MP 1900+, 768MB memory, 256 nodes) 上でシミュレーションを実行

Tierモデルでのスケジューリングと複製手法の組み合わせ(5通り)

スケジューリング手法	データ複製手法	ComputeHost 選択	複製のタイミング	複製を作るデータ	複製の送信先
Greedy	-	MCT	Read	入力データ	Compute Host
Owner Computes	-	Owner + MCT	-	-	-
LoadBound -Read	-	MCT + Load	Read	入力データ	Compute Host
LoadBound -Write	-	MCT + Load	Write	出力データ	Arbitrary
Owner Computes	LoadBound-Replication	MCT	Periodic	Arbitrary	Arbitrary

スケジューリング+オンデマンド複製 vs.
 OwnerComputes+バックグラウンド複製

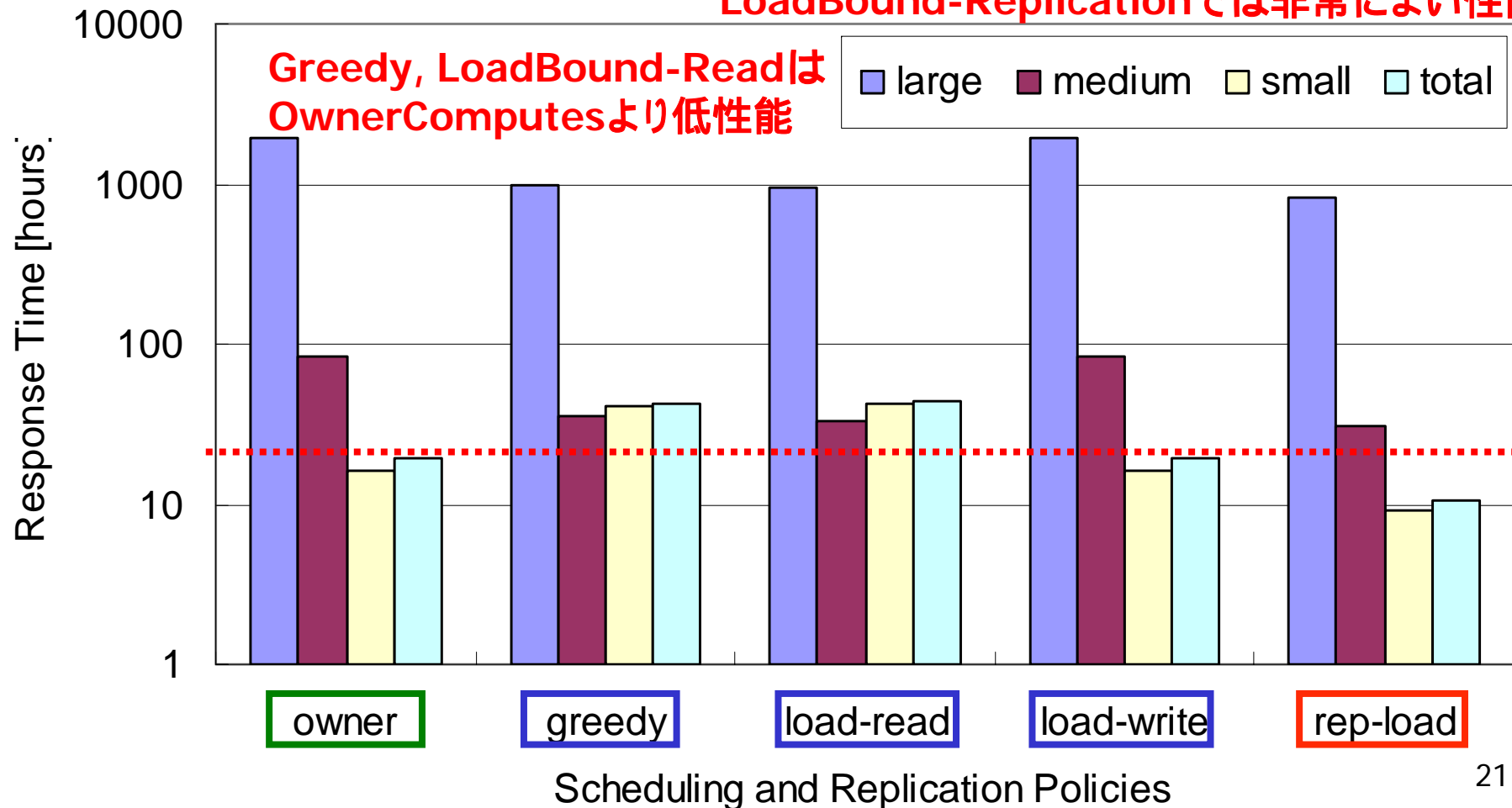
Centralモデル平均応答時間



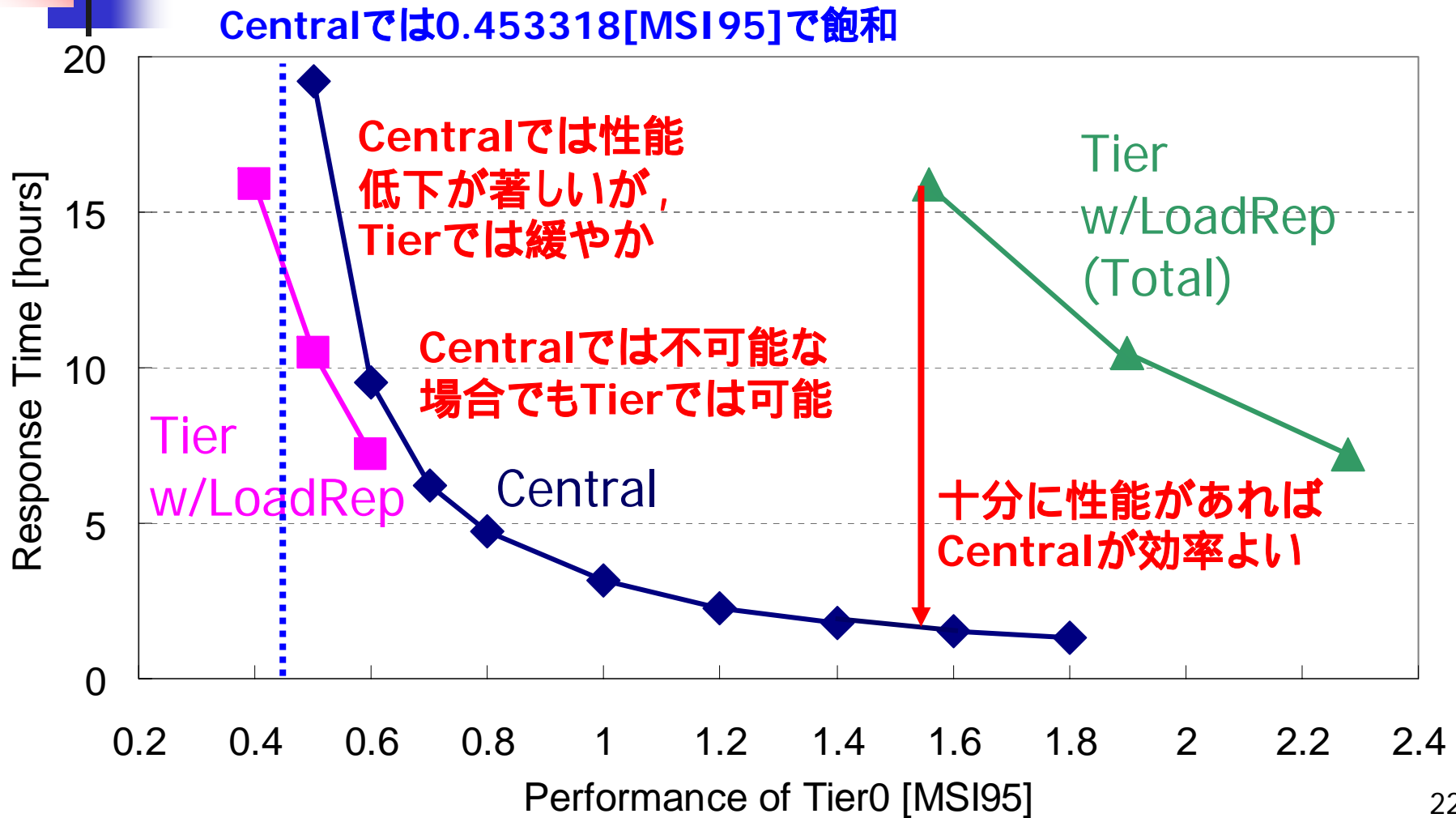
Tierモデル平均応答時間

$$(T_0, T_1, T_2) = (0.5, 0.25, 0.025)$$

OwnerComputes+
LoadBound-Replicationでは非常によい性能



CentralモデルとTierモデルの 平均応答時間の比較





関連研究: GriPhyNのシミュレーションによる評価[HPDC-11, '02]

- GriPhyN [Univ. of Chicago et al]
 - CERN LHC実験解析がターゲット
 - Globusベースのペタスケール仮想データグリッド構築
- シミュレーションによるスケジューリングと複製手法の評価
 - 外部スケジューラ, ローカルスケジューラ, データセットスケジューラによるシステムモデルを提案
 - 外部スケジューリングとデータセットスケジューリング手法の評価
 - JobDataPresent (OwnerComputes) + 複製手法でよい性能
 - LHC実験パラメータでない
(ジョブ粒度小, 短期間, データの増加なし)

→Grid Datafarmアーキテクチャを対象
LHC実験パラメータを利用した評価



まとめと今後の課題

- Bricksグリッドシミュレータをデータグリッドに拡張し, データグリッドモデルの性能を評価
- Grid Datafarmアーキテクチャを想定し, CentralモデルとMONARC型Tierモデルの比較
 - 十分な計算性能を確保できればCentralが効率よい
 - Tierでは適切なスケジューリング・複製手法 (OwnerComputes + LoadBound-Replication) により性能低下は緩やかになる
 - Centralで不可能な場合もTierで処理可能
- 今後, より効率的なスケジューリング・複製手法を提案し, 大規模環境で様々な評価していく