

Resource allocation method considering future application requests in a disaggregated micro data center

大阪大学 大学院情報科学研究科 村田研究室
 博士前期課程 2年
 生駒 昭繁

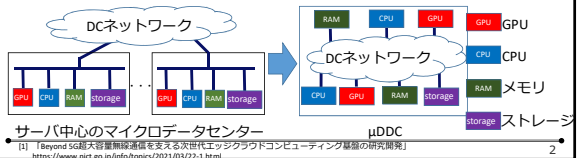
2022/2/10

令和3年度 修士論文発表会

1

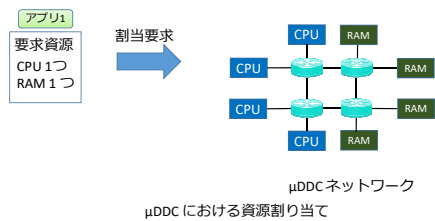
研究背景

- マイクロデータセンターを利用したエッジサービスの提供
 - エッジに配置することで通信遅延の削減を実現
 - 自動運転等、時間にセンシティブな処理に有効
 - しかし、大規模なデータセンターと比べてリソースに限りがある
- リソース分離型マイクロデータセンター (μDDC) が提案
 - リソース単位で構成されたマイクロデータセンター
 - 柔軟なスケーリングやリソース利用率の向上の実現
- 我々のプロジェクト^[1]ではマルチコアファイバを活用した高性能な μDDC を検討



μDDC における資源割り当て

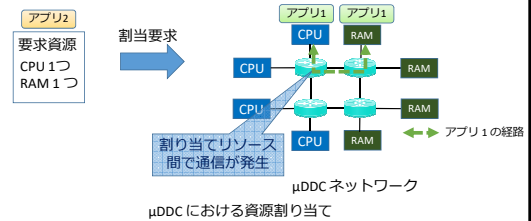
- アプリケーション開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て



3

μDDC における資源割り当て

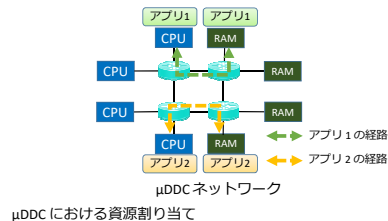
- アプリケーション開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て



3

μDDC における資源割り当て

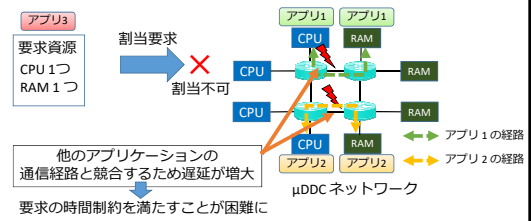
- アプリケーション開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て



3

μDDC における資源割り当て

- アプリケーション開始前に資源割り当ての要求が到着
- 要求に対して、随時、実行に必要な計算資源、メモリ資源、それらの間のネットワーク資源を割り当て



3

μDDCでは、ネットワーク資源の効率的な割り当てが必要

研究目的とアプローチ

- 研究の目的
 - リソース分離型マイクロデータセンターにおけるネットワーク資源の効率的利用による、収容可能アプリケーション数の向上
- アプローチ
 - 将来の要求収容に重要な資源を残す
 - 将来到着するアプリケーション要求に必要なとなる資源を残すことで、将来到着する性能要求の厳しいアプリケーションの収容を可能とする
 - 複数アプリケーションでのリンクの共有を許容
 - ネットワーク資源の利用効率を向上
 - アプリケーションの目標時間以内に収まる場合に共有を許容
 - ネットワーク遅延がアプリケーションの実行に与える影響をモデル化

想定する μDDC

- マルチコアファイバ (MCF) を用いたネットワークを構成
 - 内部に複数の光ファイバコアを保持
- 複数のサービスを提供
 - 継続的にサービスの割当要求が到着
- アロケータでサービス実行に利用する資源を決定
 - CPUはコア単位、メモリは一定サイズのブロック単位で割り当て
 - 同一処理を行う資源間の通信経路を決定

提案する資源割り当て

アプリケーション要求到着時に以下の手順で当該アプリケーションに必要な資源を割り当て

- 利用可能な計算機資源・メモリ資源・ネットワーク資源に対して、将来の資源割り当てへの有用性を考慮したコストを割り当て
- アプリケーション処理を目標時間内に終わらせることができる資源割り当てのうち、コスト最小の資源を割り当て

将来の資源割り当て要求に必要なとなる可能性の高い資源を残しつつ、必要なアプリケーションの処理性能を確保

割当コストの定義

高性能なCPU・大容量のメモリとその間のネットワーク資源のコストを高く設定

- CPU の割当コスト: $W_c = |c| \cdot flops_c$ (未割当コア数とFLOPSの積)
- メモリの割当コスト: $W_m = |m|$ (未割当メモリ領域の大きさ)
- リンクの割当コスト:

$$W_e = \begin{cases} \sum_{c \in C^{cand}, m \in M^{cand}} \left\{ \frac{N_{c,m}^{path}(e)}{N_{c,m}^{path}} \left(\frac{W_c \cdot W_m}{N_{c,m}^{hop}} \right) \right\} & e \notin E^{alloc} \\ 0 & \text{通信経路の資源の通信ペア } e \in E^{alloc} \end{cases}$$
 となりやすさとしての重要度

- 少ないホップ数で通信可能な高コストな資源間の経路になりやすいリンクのコストを高く設定
- 経路として割り当て済みのリンクのコストは 0

C^{cand} : 割当可能 CPU コア
 M^{cand} : 割当可能メモリ
 E^{alloc} : 割当済みリンク
 $N_{c,m}^{path}$: 資源 c,m 間の最短経路数
 $N_{c,m}^{path}(e)$: 資源 c,m 間の最短経路のうちリンク e を通る経路数
 $N_{c,m}^{hop}$: 資源 c,m 間の最短ホップ数

アプリケーション実行時間

- CPU 内処理時間とメモリからのデータ読み込みにかかる通信遅延の和として導出

$$CPU \text{ 内処理時間} = \frac{\sigma_{c,p}^c}{F_c} + \left(\frac{\sigma_{c,p}^{m,c}}{B} + S_p + T_{latency}^{c,m,p} \right) \cdot \sigma_p^{p,c} \cdot \text{通信発生回数}$$

データ読み込みにかかる通信遅延

実行に必要なクロック数をクロック周波数で除算

伝送遅延 + スイッチ処理遅延 + 伝播遅延

他のアプリケーションの通信時衝突回避のためのバッファリング遅延が発生

M/D/C 待ち行列モデルによってバッファリング時間を導出

メモリデータ読み込み時の遅延時間例

評価環境

- 2D トーラスネットワークでの割当処理をシミュレーション
 - 性能に大きく影響する CPU とメモリ、その通信経路を割り当て

評価用ネットワーク

- 3 種類の時間制約の要求が 300 分間到着
 - 1 分単位で要求が随時到着
 - 到着した要求を随時割り当て
 - 要求の寿命は 60分

要求	時間制約
要求 1	3000ms
要求 2	500ms
要求 3	150ms

評価方法

- 要求列を各要求の到着確率をもとに3通り生成

	要求1の到着確率	要求2の到着確率	要求3の到着確率
要求列1	80%	10%	10%
要求列2	40%	20%	40%
要求列3	10%	10%	80%

- 要求の必要資源数を3通りに変更して評価

	要求1の必要資源	要求2の必要資源	要求3の必要資源
必要資源数1	CPU: 7 メモリ: 7	CPU: 4 メモリ: 4	CPU: 4 メモリ: 4
必要資源数2	CPU: 4 メモリ: 4	CPU: 4 メモリ: 4	CPU: 4 メモリ: 4
必要資源数3	CPU: 4 メモリ: 4	CPU: 4 メモリ: 4	CPU: 7 メモリ: 7

- 要求列と必要資源9通りの組み合わせについて棄却数を評価
 - 時間制約を満たす資源と経路の組み合わせがない場合に棄却
 - 棄却が少ないほど収容能力が高い

10

割当方針の評価

- 将来の要求を考慮しない割当手法と棄却数を比較
 - 利用帯域が少ないネットワーク資源を優先的に割り当て
- 全環境において提案手法では棄却が発生しなかった
 - 重要な資源を残すことで厳しい制約の要求にも対応
- 比較手法では一部の環境で棄却が発生
 - 時間制約 150ms の厳しい制約の要求の資源量が多い環境
 - 全要求の 10% 以上が棄却

11

まとめと今後の課題

- まとめ
 - μDDC における資源のコストに基づく資源割当手法を提案
 - 重要な資源のコストを高く設定
 - 将来の要求に備えて資源をまとめて残す
 - リンクの共有によるネットワーク資源の効率的利用
 - アプリケーションの実行時間をモデル化
 - 提案手法により μDDC のアプリケーション収容能力が向上
- 今後の課題
 - 大規模なネットワーク環境での提案手法の評価
 - μDDC に適したネットワーク構造の検討
 - ネットワークトポロジ、リソースの配置方法

12