

リソース分離型データセンターにおける処理負荷やネットワークの変化を考慮した動的資源管理手法

大阪大学 大学院情報科学研究科 村田研究室
 博士後期課程 3年
 生駒 昭繁

2024/09/05

1

リソース分離型データセンター

- リソース分離型マイクロデータセンター (DDC) が提案
 - 資源単位で構成されたデータセンター
 - CPU やメモリがサーバから独立し、ネットワーク接続
 - サービスに必要な分だけの資源を割当可能
 - 柔軟なスケーリングや資源利用率の向上の実現
- 実行資源の柔軟な割当により、その時のサービス実行のために最適な実行環境を動的に構成可能

【従来】サーバ中心のアーキテクチャ リソース分離型アーキテクチャ

2

DDC を用いたサービス基盤

- エンド端末に対して、サービスを提供するDDC を構成
- サービスを受ける端末の増減に応じて動的に資源を割当
 - リソース分離による柔軟な資源管理で実現
 - 実行資源を動的に割当/解放
- 効率的に資源を利用しながら、負荷が変化してもサービスの要件を満たし続けることができる基盤を構築

3

DDC におけるタスクの実行

- タスク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、それらの間の経路を割当
- 計算資源とメモリ資源は必要なデータの送受信のために通信しながら処理を実行

4

DDC におけるタスクの実行

- タスク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、それらの間の経路を割当
- 計算資源とメモリ資源は必要なデータの送受信のために通信しながら処理を実行

5

DDC におけるタスクの実行

- タスク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、それらの間の経路を割当
- 計算資源とメモリ資源は必要なデータの送受信のために通信しながら処理を実行

6

DDC における資源追加の課題

- サービスの実行を要求する端末が増加
 - 実行資源にかかる負荷が増加
 - 負荷の分散のためにサービスの実行資源を追加割当

サービスの需要増により、資源の負荷が増加

サービス1の増強資源
CPU 1つ
RAM 1つ

サービス1の
実行資源として、
資源を追加割当

割当要求

タスク1の経路

タスク2の経路

7

DDC における資源追加の課題

- サービスの実行を要求する端末が増加
 - 実行資源にかかる負荷が増加
 - 負荷の分散のためにサービスの実行資源を追加割当

サービスの需要増により、資源の負荷が増加

サービス1の増強資源
CPU 1つ
RAM 1つ

割当要求

近接する空き資源を利用する場合他のタスクの通信経路と競合するため遅延が増大

タスク1の経路

タスク2の経路

サービスの時間制約を満たすことが困難になる可能性

8

7

8

DDC における資源追加の課題

- サービスの実行を要求する端末が増加
 - 実行資源にかかる負荷が増加
 - 負荷の分散のためにサービスの実行資源を追加割当

サービスの需要増により、資源の負荷が増加

サービス1の増強資源
CPU 1つ
RAM 1つ

割当要求

通信量の多いリンクを避けた経路をとるとホップ数が増大

タスク1の経路

タスク2の経路

サービスの時間制約を満たすことが困難になる可能性

9

9

DDC における動的資源管理の課題と対応

- 資源間の通信遅延がサービスの性能に影響
 - 性能要件を満たすことができない可能性がある
 - 実行サービスの性能を意識した資源管理が必要
- ある資源の割当が他資源間の通信や資源割当を制限
 - 他の実行資源間の実行を阻害しない資源管理が必要
 - サービスの実行に必要な資源の割当は避ける

↓

- 実行資源の動的な変化が性能に及ぼす影響の考慮
- 資源の追加割当が必要なサービスの予測
- 追加資源として利用される可能性の高い資源の特定

10

10

DDC における動的資源管理手法の検討

- 研究の目的
 - DDC において、割当サービスの実行要求量の変化に対する動的な資源管理手法の提案
- アプローチ
 - 割当サービスの需要予測に基づいた資源の需要を定義
 - 資源の需要をもとに資源の割当コストを拡張
 - サービスに対する実行要求量と割当資源の変化が性能に及ぼす影響をモデル化
 - 各リンクのトラフィック・要求量に応じた割当資源の増減手法を提案

11

11

負荷の変化に対応した動的資源管理

- ネットワークにかかる負荷を監視
 - 各リンクの packets 到着率 / サービスに対する実行要求の到着率
- 要求の実行待ち時間、資源間の通信遅延を導出
- 導出した遅延時間と閾値をもとに実行資源を動的に管理
 - 実行待ち時間の増加 → 計算資源の増強
 - 資源間通信遅延の増加 → 資源間経路の再割当

計算資源で実行情報を計測しアロケータ、予測器へ計測情報を送信

資源モニタ 資源情報送信

監視情報送信

アロケータ 予測提供

サービス1 提供サービス

サービスn

サービスの要求量をもとにサービスの需要予測

資源の追加割当/解放とこの判定

資源管理モジュール

12

12

サービス・資源の需要定義

- 需要の高いサービスが資源の追加割当や経路再割当時に利用されやすい資源・リンクの割当を避ける

• サービスの需要: $\eta_{s,t} = \frac{\max_{0 \leq \Delta t \leq T^d} \widehat{U}_{s,t+\Delta t}}{|C_s^v|}$

- 各サービスに対する資源の需要

• 計算資源: $\kappa_{s,c}^c = \frac{1}{\sum_{n \in M_s^v} H(\delta_n^R, c)} \cdot F_c \cdot L_c$

• メモリ資源: $\kappa_{s,m}^m = \left\{ \begin{array}{l} \text{実行中の資源と通信する可能性のある種別の} \\ \text{最短経路上のリンクとなりやすい} \\ \text{リンクの需要を高くする} \end{array} \right.$

• リンク: $\kappa_{s,e} = \frac{\sum_{a,b \in R^v} \theta(a,b,e)}{\lambda_{n_s^e,e} + \lambda_{n_d^e,e}}$

実行時間定式化

- 各マイクロサービスの実行時間とキュー内の実行待ち時間の和として定式化

- マイクロサービスの実行時間

- 計算資源内の実行時間と計算-メモリ資源間の通信遅延の総和
- データ転送時間: $\text{リンクの伝搬遅延} + \text{スイッチング遅延}$

(データサイズ/帯域幅) $\frac{D}{B} + \sum_{e \in \delta_s^P} T^L(e, \lambda^P, n_e^s)$

- 計算資源内の実行時間:

μ サービスの実行に必要なクロック数 $\frac{\Lambda_\mu^c}{F}$ 実行計算資源のクロック周波数

- 実行待ち時間

- M/M/C 待ち行列モデルによって定式化
- C=実行計算資源数として導出

$$\left(\frac{J}{D} - \lambda\right)^{-1} \left(1 + \left(1 - \frac{\lambda D}{J}\right) \left(\frac{J}{\lambda D}\right)^J \sum_{k=0}^{J-1} \frac{(\lambda D)^k}{k!}\right)^{-1}$$

資源割当問題の定義

- 性能要件を満たしながら割当資源のコストの総和を最小化することを目的とした最適化問題を定義

- 制約条件:

$$\forall s \in S, \forall n \in N_s^v \quad \sum_{s' \in S} \sum_{n' \in N^{v'}} 1_{\delta_n^R = \delta_{n'}^R} = 1$$

$$\forall s \in S, \forall c \in C_s^v \quad \delta_c^R \in C^s$$

$$\forall s \in S, \forall m \in M_s^v \quad \delta_m^R \in M^s$$

$$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R_{\delta_n^R, \delta_{n_d}^R}$$

$$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta_s^P} T^L(e, \lambda_{n_s^e, e}, n_e^s) \leq L_{e'}$$

$$\forall s \in S, \forall \mu \in N_s^l \quad T^Q(U_{\mu, t}, C_\mu, T_\mu^A) \leq W_\mu$$

$$\forall s \in S \quad T(s) \leq T_s^t$$

- 目的関数:

$$\text{minimize} \quad \sum_{c \in C_s^v} C_{\delta_c^R, t} + \sum_{m \in M_s^v} M_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta_s^P} \mathcal{E}_{e, t}$$

➢最適化問題を解くことで実行資源や解放資源を選択

資源割当問題の定義

- 性能要件を満たしながら割当資源のコストの総和を最小化することを目的とした最適化問題を定義

- 制約条件:

$$\forall s \in S, \forall n \in N_s^v \quad \sum_{s' \in S} \sum_{n' \in N^{v'}} 1_{\delta_n^R = \delta_{n'}^R} = 1$$

$$\forall s \in S, \forall c \in C_s^v \quad \delta_c^R \in C^s$$

$$\forall s \in S, \forall m \in M_s^v \quad \delta_m^R \in M^s$$

$$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R_{\delta_n^R, \delta_{n_d}^R}$$

$$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta_s^P} T^L(e, \lambda_{n_s^e, e}, n_e^s) \leq L_{e'}$$

$$\forall s \in S, \forall \mu \in N_s^l \quad T^Q(U_{\mu, t}, C_\mu, T_\mu^A) \leq W_\mu$$

$$\forall s \in S \quad T(s) \leq T_s^t$$

- 目的関数:

$$\text{minimize} \quad \sum_{c \in C_s^v} C_{\delta_c^R, t} + \sum_{m \in M_s^v} M_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta_s^P} \mathcal{E}_{e, t}$$

➢最適化問題を解くことで実行資源や解放資源を選択

資源割当問題の定義

- 性能要件を満たしながら割当資源のコストの総和を最小化することを目的とした最適化問題を定義

- 制約条件:

$$\forall s \in S, \forall n \in N_s^v \quad \sum_{s' \in S} \sum_{n' \in N^{v'}} 1_{\delta_n^R = \delta_{n'}^R} = 1$$

$$\forall s \in S, \forall c \in C_s^v \quad \delta_c^R \in C^s$$

$$\forall s \in S, \forall m \in M_s^v \quad \delta_m^R \in M^s$$

$$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R_{\delta_n^R, \delta_{n_d}^R}$$

$$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta_s^P} T^L(e, \lambda_{n_s^e, e}, n_e^s) \leq L_{e'} \quad \text{実行資源間の通信遅延が閾値以下}$$

$$\forall s \in S, \forall \mu \in N_s^l \quad T^Q(U_{\mu, t}, C_\mu, T_\mu^A) \leq W_\mu$$

$$\forall s \in S \quad T(s) \leq T_s^t$$

- 目的関数:

$$\text{minimize} \quad \sum_{c \in C_s^v} C_{\delta_c^R, t} + \sum_{m \in M_s^v} M_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta_s^P} \mathcal{E}_{e, t}$$

➢最適化問題を解くことで実行資源や解放資源を選択

資源割当問題の定義

- 性能要件を満たしながら割当資源のコストの総和を最小化することを目的とした最適化問題を定義

- 制約条件:

$$\forall s \in S, \forall n \in N_s^v \quad \sum_{s' \in S} \sum_{n' \in N^{v'}} 1_{\delta_n^R = \delta_{n'}^R} = 1$$

$$\forall s \in S, \forall c \in C_s^v \quad \delta_c^R \in C^s$$

$$\forall s \in S, \forall m \in M_s^v \quad \delta_m^R \in M^s$$

$$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R_{\delta_n^R, \delta_{n_d}^R}$$

$$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta_s^P} T^L(e, \lambda_{n_s^e, e}, n_e^s) \leq L_{e'} \quad \mu\text{サービスの実行待ち時間が閾値以下}$$

$$\forall s \in S, \forall \mu \in N_s^l \quad T^Q(U_{\mu, t}, C_\mu, T_\mu^A) \leq W_\mu$$

$$\forall s \in S \quad T(s) \leq T_s^t$$

- 目的関数:

$$\text{minimize} \quad \sum_{c \in C_s^v} C_{\delta_c^R, t} + \sum_{m \in M_s^v} M_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta_s^P} \mathcal{E}_{e, t}$$

➢最適化問題を解くことで実行資源や解放資源を選択

資源割当問題の定義

- 性能要件を満たしながら割当資源のコストの総和を最小化することを目的とした最適化問題を定義

- 制約条件:

$$\forall s \in S, \forall n \in N_s^v \quad \sum_{e' \in E_{s'}} \sum_{n' \in N_{n'}^v} \lambda_{\delta_{n'}^R} = 1$$

$$\forall s \in S, \forall c \in C_s^v \quad \delta_c^R \in C^s$$

$$\forall s \in S, \forall m \in M_s^v \quad \delta_m^R \in M^s$$

$$\forall s \in S, \forall e \in E_s^v \quad \delta_e^R \in R_{\delta_{n_e^R}^s, \delta_{n_e^R}^R}$$

$$\forall s \in S, \forall e' \in E_{s'}^v \quad \sum_{e \in \delta_{e'}^R} T^L(e, \lambda_{n_e^s, e}, n_e^s) \leq L_{e'}$$

$$\forall s \in S, \forall \mu \in N_{\mu}^l \quad T^Q(U_{\mu, t}, C_{\mu}, T_{\mu}^A) \leq W_{\mu}$$

$$\forall s \in S \quad T(s) \leq T_s^l$$

- 目的関数:

$$\text{minimize} \quad \sum_{c \in C_s^v} C_{\delta_{c,t}^R} + \sum_{m \in M_s^v} M_{\delta_{m,t}^R} + \sum_{e' \in E_{s'}^v} \sum_{e \in \delta_{e'}^R} \mathcal{E}_{e,t}$$

➢最適化問題を解くことで実行資源や解放資源を選択

資源割当問題の定義

- 性能要件を満たしながら割当資源のコストの総和を最小化することを目的とした最適化問題を定義

- 制約条件:

$$\forall s \in S, \forall n \in N_s^v \quad \sum_{e' \in E_{s'}} \sum_{n' \in N_{n'}^v} \lambda_{\delta_{n'}^R} = 1$$

$$\forall s \in S, \forall c \in C_s^v \quad \delta_c^R \in C^s$$

$$\forall s \in S, \forall m \in M_s^v \quad \delta_m^R \in M^s$$

$$\forall s \in S, \forall e \in E_s^v \quad \delta_e^R \in R_{\delta_{n_e^R}^s, \delta_{n_e^R}^R}$$

$$\forall s \in S, \forall e' \in E_{s'}^v \quad \sum_{e \in \delta_{e'}^R} T^L(e, \lambda_{n_e^s, e}, n_e^s) \leq L_{e'}$$

$$\forall s \in S, \forall \mu \in N_{\mu}^l \quad T^Q(U_{\mu, t}, C_{\mu}, T_{\mu}^A) \leq W_{\mu}$$

$$\forall s \in S \quad T(s) \leq T_s^l$$

- 目的関数:

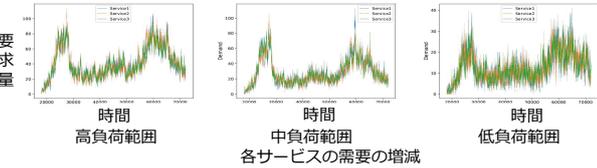
$$\text{minimize} \quad \sum_{c \in C_s^v} C_{\delta_{c,t}^R} + \sum_{m \in M_s^v} M_{\delta_{m,t}^R} + \sum_{e' \in E_{s'}^v} \sum_{e \in \delta_{e'}^R} \mathcal{E}_{e,t}$$

➢最適化問題を解くことで実行資源や解放資源を選択

サービスの割当資源のコストの総和が最小となるように資源を割当/解放

シミュレーション評価

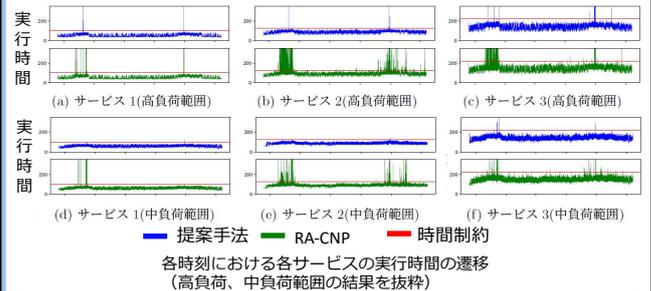
- 3種のサービスが提供される DDC を想定
- 24時間のドイツのケルン市の1秒ごとの車両位置データ[1]をもとに各サービスの要求を生成
 - 車両通行量が異なる3範囲で評価
- Temporal Fusion Transformerモデルにより要求量予測
- 新規サービス要求に有効な資源を多く残す手法(RA-CNP)[2]と比較
 - 提案手法と異なり、実行資源との新規割当資源との接続は考慮しない
 - 私たちが以前提案した DDC に向けた資源割当手法



[1] S. Uppoor, O. Trullols-Cruces, M. Fiore, and J. M. Barcelo-Ordinas, "Generation and analysis of a large-scale urban vehicular mobility dataset," IEEE Transactions on Mobile Computing, vol. 13, no. 5, pp. 1061–1075, 2014.
 [2] Akehiyo Ikoma, Y. Ohtsuki, and M. Murata, "Resource Allocation Considering Impact of Network on Performance in a Disaggregated Data Center," in IEEE Access, vol. 12, no. 67600–67618, May 2024.

評価結果

- 各時刻におけるアプリケーションの実行時間を計測
- 比較手法と比較して制約を満たさない期間を大きく抑制
- 負荷の変化に柔軟に対応可能な資源管理を実現



まとめと今後の予定

- まとめ
 - DDCにおけるサービスの要求量に基づく動的資源管理を提案
 - 負荷の変化に基づき資源の追加割当や経路再割り当てを実施
 - 資源割当時、将来の需要が高い資源を温存
 - 負荷が動的に変動する環境において、より長い時間サービスの性能要件を満たし続けることに成功
- 今後の予定
 - 通信する資源への種類を問わない形での定式化
 - 手法の計算時間の評価

資源割当コストの定義

- DDCで実行されるサービスの需要と各サービスに対する資源の需要の積の総和として各資源の割当コストを定義
- コストが高いほど追加資源として割り当てられやすい
 - 需要が高いサービスほど負荷が高まる可能性が高い

- 計算資源

$$C_{c,t} = \sum_{s \in S} \{ \eta_{s,t} \cdot \gamma_{s,c} \}$$

$$M_{m,t} = \sum_{s \in S} \{ \eta_{s,t} \cdot \zeta_{s,m} \}$$

$$\mathcal{E}_{e,t} = \sum_{s \in S} \{ \eta_{s,t} \cdot \kappa_{s,e} \}$$