Dynamic Resource Allocation Considering Workload Changes in a Disaggregated Data Center

Akishige Ikoma, Yuichi Ohsita, Masayuki Murata
Graduate School of Information Science and Technology, Osaka University

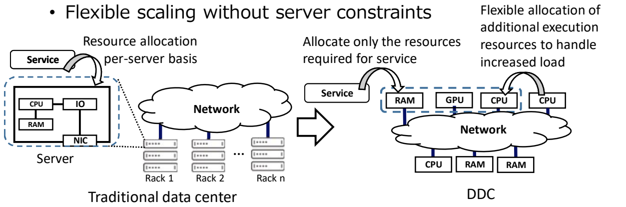
2025/01/06 1

1

Background

- Edge services using micro data centers
  - Smaller latency than the cloud
    - Effective for time-sensitive services
  - More limited resources compared with large data centers

➢Disaggregated Micro Data Center (DDC)
  - DDC is constructed of resources connected by a network
  - Achieve efficient resource utilization
    - Optimization per resource
  - Flexible scaling without server constraints

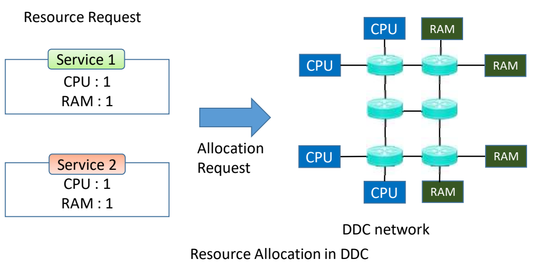


2

2

Service execution in DDC

- Resource request arrives before service starts
- Allocate execution computational resources, memory resources, and routes between resources
- Computational and memory resources execute services while communicating

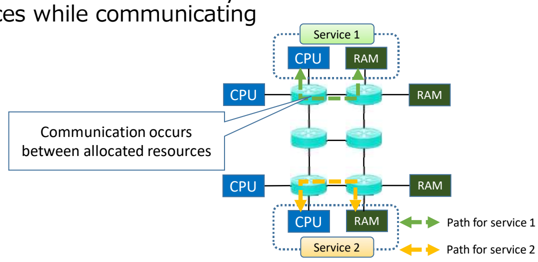


3

3

Service execution in DDC

- Resource request arrives before service starts
- Allocate execution computational resources, memory resources, and routes between resources
- Computational and memory resources execute services while communicating

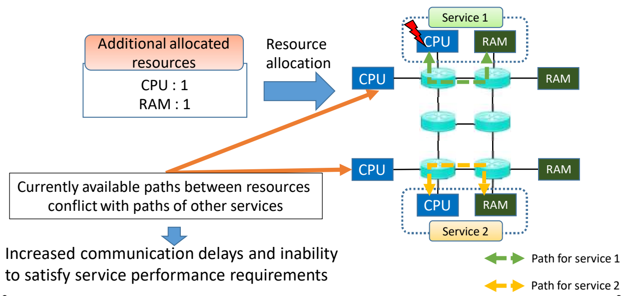


4

4

Problem of workload changes in DDC

- Workload changes due to changing service demand
  - Increase processing volume of computing resources
  - Increase in communication traffic
- Allocate additional execution resources for balancing

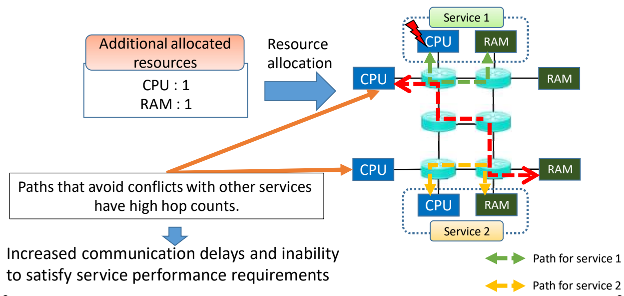


5

5

Problem of workload changes in DDC

- Workload changes due to changing service demand
  - Increase processing volume of computing resources
  - Increase in communication traffic
- Allocate additional execution resources for balancing



6

6

Resource management for DDC

- Resource communication affects performance
  - May not satisfy performance requirements
  - ➢ Resource management considering performance is essential
- Resource allocation affects other resource allocation
  - Resource management that does not interfere with other service resource allocations is necessary

Resource management

- Consideration of the impact of allocated resources on performance
- Prediction of services that require additional resource allocation
- Identification of resources likely to be used as additional resources

7

7

Approach

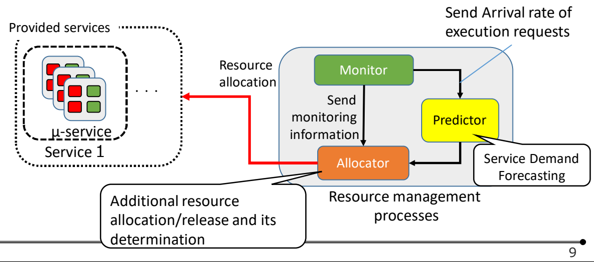
- Research objective
  - Continually satisfy the performance requirements for service in a DDC where workloads change dynamically
- Approach
  - Define demand of resources based on future services demand
    - Leave high demand resources for future resource allocation
  - Model the impact of workload and allocated resources on performance for services
    - Enables resource allocation to satisfy performance requirements

8

8

Resource management considering workload change

1. Monitoring workload in DDC
   - Arrival rate of execution requests for service (service demand)
   - Traffic on each network link
2. Derive execution waiting time and communication delay between resources
3. Dynamic resource allocation based on threshold



9

9

Resource allocation method (DRA-CWC)

- Avoid allocating resources/links that are likely to be allocated for services with high demand

1. Formulate the demand for each service
2. Formulate the demand for each resource and link for the executing services
3. Cost assign to resources
   - Based on the demands of services and resources
4. Determine resource allocation that minimizes cost while satisfying service performance requirements

10

10

Demand for services and resources

- Based on the number of service execution requests and resource performance and proximity
- Service demand $\eta_{s,t} = \frac{\max_{0 \le \Delta t \le T^d} \widehat{U}_{s,t+\Delta t}}{|C_s^v|}$
- Demand of resources for the executing services
  - Computational resource : $\kappa_{s,c}^c = \frac{1}{\sum_{a \in M_s^v} H(\delta_a^R, c)} \cdot F_c \cdot L_c$
  - Memory resource : $\kappa_{s,m}^m = \left\{ \frac{1}{\sum_{a \in M_s^v} H(m, \delta_a^R)} + \frac{1}{\sum_{b \in C^s} H(m, b)} \right\} \cdot L_m$
  - Link : $\kappa_{s,e} = \frac{\sum_{a,b \in R^p} \theta(a,b,e)}{\lambda_{n_e^s,e} + \lambda_{n_e^d,e}}$

11

11

Demand for services and resources

- Based on the number of service execution requests and resource performance and proximity
- Service demand $\eta_{s,t} = \frac{\max_{0 \le \Delta t \le T^d} \widehat{U}_{s,t+\Delta t}}{|C_s^v|}$

Ratio of the number of allocated resources to the number of future service execution requests

- Demand of resources
  - Computational resource : $\kappa_{s,c}^c = \frac{1}{\sum_{a \in M_s^v} H(\delta_a^R, c)} \cdot F_c \cdot L_c$
  - Memory resource : $\kappa_{s,m}^m = \left\{ \frac{1}{\sum_{a \in M_s^v} H(m, \delta_a^R)} + \frac{1}{\sum_{b \in C^s} H(m, b)} \right\} \cdot L_m$
  - Link : $\kappa_{s,e} = \frac{\sum_{a,b \in R^p} \theta(a,b,e)}{\lambda_{n_e^s,e} + \lambda_{n_e^d,e}}$

12

12

Demand for services and resources

- Based on the number of service execution requests and resource performance and proximity
- Service demand $\eta_{s,t} = \frac{\max_{0\le\Delta t\le T^d} \widehat{U}_{s,t+\Delta t}}{|C^v_s|}$
- Demand of resources for the executing services
  - Computational resource : $\kappa^c_{s,c} = \frac{1}{\sum_{a\in M^v_s} H(\delta^R_a, c)} \cdot F_c \cdot L_c$
  - Memory resource : $\kappa^m_{s,m}$ …
  - Link : $\kappa_{s,e} = \frac{\sum_{a,b\in R^p} \theta(a,b,e)}{\lambda_{n^s_e,e} + \lambda_{n^d_e,e}}$

High performance resources in close proximity to allocated memory resources are high demand

13

Demand for services and resources

- Based on the number of service execution requests and resource performance and proximity
- Service demand $\eta_{s,t} = \frac{\max_{0\le\Delta t\le T^d} \widehat{U}_{s,t+\Delta t}}{|C^v_s|}$
- Demand of resources for the executing services
  - Computational resource : $\kappa^c_{s,c} = \frac{1}{\sum_{a\in M^v_s} H(\delta^R_a, c)} \cdot F_c \cdot L_c$
  - Memory resource : $\kappa^m_{s,m} = \left\{\frac{1}{\sum_{a\in M^v_s} H(m,\delta^R_a)} + \frac{1}{\sum_{b\in C^s} H(m,b)}\right\} \cdot L_m$
  - Link : $\kappa_{s,e} = \frac{\sum_{a,b} \dots}{\lambda_{n \dots}}$

High performance resources in close proximity to allocated memory and computational resources are high demand

14

Demand for services and resources

- Based on the number of service execution requests and resource performance and proximity
- Service demand $\eta_{s,t} = \frac{\max_{0\le\Delta t\le T^d} \widehat{U}_{s,t+\Delta t}}{|C^v_s|}$
- Demand of resources for the executing services
  - Computational resource : $\kappa^c_{s,c} = \frac{1}{\sum_{a\in M^v_s} H(\delta^R_a, c)} \cdot F_c \cdot L_c$
  - Memory resourc…
  - Link : $\kappa_{s,e} = \frac{\sum_{a,b\in R^p} \theta(a,b,e)}{\lambda_{n^s_e,e} + \lambda_{n^d_e,e}}$

Links that are likely to be on the shortest path between resources are in high demand

15

Model of execution time

- Sum of processing time and execution waiting time
- Processing time of service
  - Communication delay :

$$\frac{D}{B} + \sum_{e\in\delta^P_{e'}} T^L(e, \lambda^P, n^s_e)$$

transmission delay (data size / bandwidth)

Propagation delay + Switching delay

  - Processing time in computational resource :

$$\frac{\Lambda^c_\mu}{F}$$

the number of clocks to execute service

Clock frequency of computational resource

- Execution waiting time
  - Based on M/M/C queue
    - J = the number of allocated computational resources
    - λ = Arrival rate of execution requests for service
    - D = Processing time of service

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

16

Resource Allocation Problem

- Minimize cost while satisfying performance requirements of services
- Constraints :

$$\forall s\in S, \forall n\in N^v_s \quad \sum_{s'\in S}\sum_{n'\in N^v} 1_{\delta^R_n=\delta^R_{n'}} = 1$$
$$\forall s\in S, \forall c\in C^v_s \quad \delta^R_c \in C^s$$
$$\forall s\in S, \forall m\in M^v_s \quad \delta^R_m \in M^s$$
$$\forall s\in S, \forall e\in E^v_s \quad \delta^P_e \in R^s_{\delta^R_{n^s_e},\delta^R_{n^d_e}}$$
$$\forall s\in S, \forall e'\in E^v_s \quad \sum_{e\in\delta^P_{e'}} T^L(e,\lambda_{n^s_e,e}, n^s_e) \le \mathbb{L}_{e'}$$
$$\forall s\in S. \forall u\in N^\mu_s \quad T^Q(U_{\mu,t}, C_\mu, T^A_\mu) \le \mathbb{W}_\mu$$
$$\forall s\in S \quad T(s) \le T^t_s$$

- Objective :

$$minimize \quad \sum_{c\in C^v_s} \mathcal{C}_{\delta^R_c,t} + \sum_{m\in M^v_s} \mathcal{M}_{\delta^R_m,t} + \sum_{e'\in E^v_s}\sum_{e\in\delta^P_{e'}} \mathcal{E}_{e,t}$$

- Derived by metaheuristic method for NP-hard
  - We use Ant Colony Optimization (ACO)

17

Resource Allocation Problem

- Minimize cost while satisfying performance requirements of services
- Constraints :

$$\forall s\in S, \forall n\in N^v_s \quad \sum_{s'\in S}\sum_{n'\in N^v} 1_{\delta^R_n=\delta^R_{n'}} = 1$$
$$\forall s\in S, \forall c\in C^v_s \quad \delta^R_c \in C^s$$
$$\forall s\in S, \forall m\in M^v_s \quad \delta^R_m \in M^s$$
$$\forall s\in S, \forall e\in E^v_s \quad \delta^P_e \in R^s_{\delta^R_{n^s},\delta^R_{n^d}}$$

  - Request resources and resources are one-to-one
  - No more than one service can be allocated to one resource

$$\forall s\in S. \forall u\in N^\mu_s \quad T^Q(U_{\mu,t}, C_\mu, T^A_\mu) \le \mathbb{W}_\mu$$
$$\forall s\in S \quad T(s) \le T^t_s$$

- Objective :

$$minimize \quad \sum_{c\in C^v_s} \mathcal{C}_{\delta^R_c,t} + \sum_{m\in M^v_s} \mathcal{M}_{\delta^R_m,t} + \sum_{e'\in E^v_s}\sum_{e\in\delta^P_{e'}} \mathcal{E}_{e,t}$$

- Derived by metaheuristic method for NP-hard
  - We use Ant Colony Optimization (ACO)

18

Resource Allocation Problem

- Minimize cost while satisfying performance requirements of services
- Constraints :

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

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

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

$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R^s_{\delta^R_{n_e^s}, \delta^R_{n_e^d}}$

$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta^P_{e'}} T^L(e, \lambda_{n_e^s, e}, n_e^s) \leq \mathbb{L}_{e'}$

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

- Communication delay between resources is less than threshold
- Execution waiting time is less than threshold

- Objective :

$$minimize \sum_{c \in C_s^v} \mathcal{C}_{\delta_c^R, t} + \sum_{m \in M_s^v} \mathcal{M}_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta^P_{e'}} \mathcal{E}_{e,t}$$

- Derived by metaheuristic method for NP-hard
  - We use Ant Colony Optimization (ACO)

Resource Allocation Problem

- Minimize cost while satisfying performance requirements of services
- Constraints :

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

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

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

$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R^s_{\delta^R_{n_e^s}, \delta^R_{n_e^d}}$

$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta^P_{e'}} T^L(e, \lambda_{n_e^s, e}, n_e^s) \leq \mathbb{L}_{e'}$

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

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

Finish the process within the acceptable time

- Objective :

$$minimize \sum_{c \in C_s^v} \mathcal{C}_{\delta_c^R, t} + \sum_{m \in M_s^v} \mathcal{M}_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta^P_{e'}} \mathcal{E}_{e,t}$$

- Derived by metaheuristic method for NP-hard
  - We use Ant Colony Optimization (ACO)

Resource Allocation Problem

- Minimize cost while satisfying performance requirements of services
- Constraints :

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

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

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

$\forall s \in S, \forall e \in E_s^v \quad \delta_e^P \in R^s_{\delta^R_{n_e^s}, \delta^R_{n_e^d}}$

$\forall s \in S, \forall e' \in E_s^v \quad \sum_{e \in \delta^P_{e'}} T^L(e, \lambda_{n_e^s, e}, n_e^s) \leq \mathbb{L}_{e'}$

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

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

- Objective :

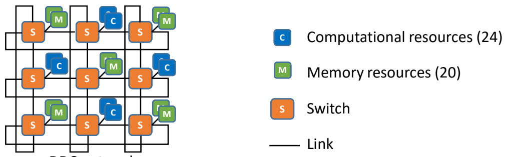
Allocate resources and paths to minimize costs

$$minimize \sum_{c \in C_s^v} \mathcal{C}_{\delta_c^R, t} + \sum_{m \in M_s^v} \mathcal{M}_{\delta_m^R, t} + \sum_{e' \in E_s^v} \sum_{e \in \delta^P_{e'}} \mathcal{E}_{e,t}$$

- Derived by metaheuristic method for NP-hard
  - We use Ant Colony Optimization (ACO)

Evaluation

- Simulate DDC with 3 services provided
- Generate execution requests for each service based on 24-hour vehicle location data set [1]
  - Evaluated in 2 ranges with different vehicle traffic
- Comparison of two resource allocation methods
  - RA-CNP: Avoid the allocation of high-performance resources and low-latency paths
  - NP: Allocates paths with low traffic and short lengths
- Measure the execution time of each service per second



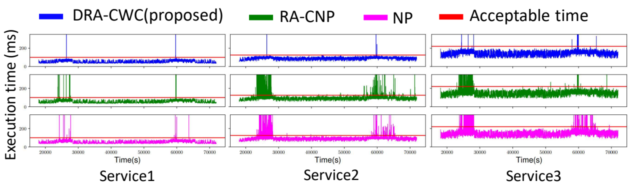
DDC network

[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.

Result

- Measure service execution time at each time
- Significantly reduces the number of periods when constraints are not satisfied

➢Enables flexible resource management in response to workload changes by DRA-CWC



Transition of execution time for each service(Selected results in high load environments)

Conclusion

- We proposed dynamic resource allocation considering workload changes DRA-CWC
  - Leave high demand resources for future resource allocation
  - Model the impact of workload and allocated resources on performance for services
- DRA-CWC can satisfy service performance requirements for longer time
- Future work
  - Consider communications between various types of resources
    - Focus on communication between computing and memory resources in this paper