Master's Thesis

Title

Network Structure for Micro Disaggregated Data Centers to Accommodate Many Applications

Supervisor

Professor Masayuki Murata

Author

Masashi Sorimachi

February 2nd, 2023

Department of Information Networking

Graduate School of Information Science and Technology

Osaka University

Master's Thesis

Network Structure for Micro Disaggregated Data Centers to Accommodate Many Applications

Masashi Sorimachi

Abstract

Edge computing is one of the promising approaches to providing time-sensitive applications. In this approach, many micro data centers (μDC) are deployed near users. Each user has a micro data center that is close to him/her and has more resources than the end devices. Thus, time-sensitive applications can also be provided by using micro data centers.

Effective use of limited resources is required to run a wide variety of applications because resources in μDC are limited. Therefore, a resource disaggregated micro data center (μDDC) has been proposed. In a μDDC, resources such as CPU, GPU, memory, and storage are disaggregated, and connected via a network. Resources can be allocated to each application only as needed.

The network in a μDDC has a significant impact on the performance of the applications running in the μDDC. Therefore, in this thesis, we discuss a suitable network structure for μDDC. We first discuss the requirements for the network in the μDDC. Then we introduce metrics that can be used to evaluate whether those requirements are met.

We investigate the network characteristics based on the metrics we introduce and discuss the suitable network structure by comparing the characteristics and the results of simulation of the resource allocation to the applications. The results show that the network structure that satisfies (1) the number of memory resources within a small number of hops from each computational resources is large and (2) weighed edge betweenness centrality focusing on the communication only between near resources is large is suitable for μDDC.

Keywords

Disaggregation

Micro data center

Network structure

Contents

List of Figures

List of Tables

1 Introduction

Numerous cloud-based services has become available. Such services are based on processing in large data centers located far from the user. So, time-sensitive applications are difficult to be provided through the cloud due to the large latency between the cloud data center and end device.

Edge computing is one of the promising approaches to providing time-sensitive applications. In this approach, many micro data centers (μDC) are deployed near users. Each user has a micro data center that is close to him/her and has more resources than the end devices. Thus, time-sensitive applications can also be provided by using micro data centers.

Effective use of limited resources is required to run a wide variety of applications because resources in μDC are limited. Therefore, a resource disaggregated micro data center (μDDC) has been proposed [1]. In a μDDC, resources such as CPU, GPU, memory, and storage are disaggregated, and connected via a network. Resources can be allocated to each application only as needed. That is, the resources can be used flexiblly. Furthermore, as resources in μDDC are independent, each resource can be replaced as technology evolves.

In a μDDC, resources communicate with each other via a network in the μDDC. The network in the μDDC has a significant impact on the performance of applications running in the μDDC; the large communication delay between the CPU and memory degrade the performance of the application using them. That is, it is necessary for μDDC to allocate resources considering the communication delay.

The network structure in a μDDC is important to realize flexible resource allocation in μDDC. The candidate paths between the resources depends on the network structure. If the network does not have enough candidate paths between resources, it is difficult to find the paths with low latency. As a result, it is also difficult to allocate resources to applications so as to achieve the required performances.

Several network architectures for a disaggregated data center have been proposed. Yuan et al. proposed a network structure based on multiple tree structures [2]. In this network structure, the number of hops between resources is small. That is, the low lantency and lo power communication can be achieved.

The suitable network structure also depends on the resource allocation. However, most of the existing papers focus only on the latency between the CPU and memory resources and do not consider the resource allocation policy.

In this thesis, we discuss the requirements considering the resource allocation and discuss the network structure suitable for the resource disaggregation. We first discuss the requirements for the network in the μDDC. Then we introduce metrics that can be used to evaluate whether those requirements are met. We investigate the network characteristics based on the metrics we introduce and discuss the suitable network structure by comparing the characteristics and the results of simulation of the resource allocation to the applications.

The rest of this thesis is organized as follows. Section 2 explains the related work. Section 3 investigate the network structures suitable for a μDDC. In Section 3, we first discuss the requirements for the network structures in a μDDC, and introduce metrics corresponding to each requirement. Then, we investigate the network structures based on our metric. In Section 4, we discuss the suitable network structures based on the results in Section 3. Finally, Section 5 conclude this thesis.

2 Related Work

2.1 Recent research on resource disaggregation

A DDC (Disaggregated Data Center) has been proposed as a data center where resource units such as CPUs, GPUs, memory and storage are connected via a network. In a DDC, resources can be allocated to each application only as needed. That is, the resources can be used flexiblly [3].

The network bandwidth required by DDCs has been investigated [4–6]. Gao et al. discussed the network requirements for resource disaggregation and concluded that high-bandwidth and low-latency communication is necessary [4]. In particular, communication delays between the CPU and remote memory can degrade performance. Zervas et al. investigated the requirements for optical transmission technology in DDCs [5]. As a result, they concluded that large DDCs running massively parallel computing applications require high-bandwidth communication devices that can support communications with more than 1 Tbps rate. Ikoma et al. investigated the performance of the application based on machine learning running on a μDDC [6]. The results showed that latency has a significant impact on application performance, while bandwidths of more than 50 Gbps are not required.

The network performance, which has a large impact on the application performance, resource allocation has an important role. Therefore, Ikoma et al. proposed a method that allocates computational, memory, and network resources so as to satisfy the application performance [7]. However, the resources that can be allocated depends on he network strucure. Therefore, we discuss a network structure suitable for μDDC in this thesis.

2.2 Network structure for disaggregated data center

Many network structures for data centers have been proposed [8–10]. However, the requirements for DDC are different from those of a conventional data center; all servers communicate with each other to handle a large amount of data in conventional data centers, while frequent communication occurs between allocated resources in a DDC.

Yuan et al. proposed a network structure based on multiple tree structures [2]. They compared their network structure with the traditional three-tier network structure constructed of core/aggregation/access switches. The results shows that their proposed net-

work structure reduces the latency by 34%, latency by 40% and power consumption by 68%, compared with a traditional three-tier network structure with the sane number of resources, links, switches, and switch ports.

Mishhra et al. proposed an architecture that enable memory accesses over optical network [11]. In this architecture, they used a parallel network structure constructed of multiple switch planes that provides full connectivity between all resources. They demonstrated that the performance degradation of their architecture compared with the architecture without disaggregation is only 10%.

The suitable network structure also depends on the resource allocation. However, most of the existing papers focus only on the latency between the CPU and memory resources and do not consider the resource allocation policy. In this thesis, we discuss the requirements considering the resource allocation and discuss the network structure suitable for the resource disaggregation.

3 Network structure in μDDCs

3.1 Requirements for network structure in μDDC

A μDDC accommodates applications by allocating computational and memory resources, and network resources between them. Then, the applications are run by using the allocated resources. Computational and memory resources frequently communicate with each other to read and write data from/to memories when running the applications. If the allocated network resources between computational and memory resources is insufficient, it takes a long time to obtain required data, which causes the performance degradation of the application. That is, the network has a large impact on the performance of the application.

Therefore, in this thesis, we discuss a network structure suitable for a μDDC. First, in this subsection, we summarize the requirements for network structures in a μDDC.

3.1.1 Requirement 1: Low latency between computational and memory resources

As described above, computational and memory resources communicates with each other during application execution in μDDC. If a computational resource does not have required information, it accesses a memory resources and waits for the required information to be obtained. That is, large latency between computational and memory resources causes long time for computational resources to obtain the required information, which degrades the performance.

3.1.2 Requirement 2: Large bandwidth between computational and memory resources

In a μDDC, multiple applications are executed simultaneously. That is, a large amount of communication between computational and memory resources are generated. The network in a μDDC is required to have enough bandwidth to accommodate such a large amount of communication.

3.1.3 Requirement 3: Low cost

In addition to the requirements related to performance, the cost to build a data center is also important.

3.2 Metrics for network structure in μDDC

In this subsection, we introduce metrics corresponding to the above requirements.

3.2.1 Metric 1: Number of memory resources within k hops from each computational resource

As discussed above, memory resources is required to be accessed with low latency from the computational resources using them. A a metric to check if this requirement is satisfied, we introduce a metric of the number of memory resources that can be accessed within k hops from computational resources. If a network has many memory resources that can be accessed within k hops from computational resources, it has many memory resources that can be accessed with low latency from computational resources.

3.2.2 Metric 2: Weighted edge betweenness centrality

As described above, a μDDC is required to provide sufficient bandwidth to accommodate many applications. The bandwidth that can be provided to the communication between resources depends on the bandwidth and number of communication on the bottleneck links.

One of the well-known metric related to the bottleneck links to edge betweenness centrality. The edge betweenness centrality is a metric indicating the concentration of communications on a link when the shortest paths are used.

Considering the policy of the resource allocation, the computational and memory resources that are close to each other are tend to be allocated to avoid large latency and consuming a large amount of network resources. Therefore, instead of the betweenness centrality, we introduce the metrics focusing on the communication between the near resources.

In this thesis, we introduce a metric called *weighted edge betweenness centrality*, defined below.

$$c\left(e,k\right)=\sum_{s,t\in V}\left(\frac{\sigma\left(s,t\left|e\right.\right)}{\sigma\left(s,t\right)}\cdot w_{s,t}\left(k\right)\right) \tag{1}$$

where $\sigma\left(s,t\right)$ is the number of shortest paths between s,t, $\sigma\left(s,t\left|e\right.\right)$ denotes the number of shortest paths between s,t through edge e. $w_{s,t}(k)$ is a value indicating whether there is communication between resource pairs at two points s,t. In this thesis, we set $w_{s,t}(k)$ to the value determined by the rank k of the resource pair as follows.

$$w_{s,t}\left(k\right)=\begin{cases} w'_{type(s)type(t)} & if\ rank_s(t)\leq k; \\ 0 & otherwise. \end{cases} \tag{2}$$

where $rank_s(t)$ indicates the rank of the number of hops between the resource pair; if t is the kth closest resources from m among the resources of the same type as t, $rank_s(t)=k$. $w'_{type(s)type(t)}$ indicates whether there is communication between the resources with types of $type(s)$ and $type(t)$ resources, and is set to 1 if communication is possible between $type(s)$ and $type(t)$ and 0 if it is impossible.

From the above definition, $c\left(e,k\right)$ indicates the probability that link e is used when each resource communicates only with the k resources. That is, the links with large $c\left(e,k\right)$ may becomes bottlenecks. Therefore, by checking the maximum of $c\left(e,k\right)$, we can check if a bottleneck link to provide enough bandwidth exists

3.2.3 Metric 3: Switch cost Link cost

In this thesis, we compare the cost to construct a network by two metrics the number of switches per compute and memory resource (switch cost) and the number of links (link cost).

3.3 Investigation on existing network structures

In this section, we investigate the existing network structure for data centers by using the metrics defined above. We also investigate the number of application that can be accommodated by using the simulation of the resource allocation.

3.3.1 Network structures

In this section, we investigate the characteristics of the network structures. Network structures are constructed of two parts, the connections between switches and the connection from the resources to switches. As the connections between switches, we use the three types of the network stractures that are commoly used in HPC environemtns and data centers; Torus, HyperX [8], and Fat-tree [9]. As the connections from the resources to the switches, we use two policies; (1) Alternate arrangement: the switches adjacent to the switches connected to the computing resources are connected to the memory resources, and (2) Clustered arrangement: the switches adjacent to the switches connected to the computing resources tend to be connected to the computing resources. In the alternate arrangement, each computational resource can access the memory resources with a small number of hops.

In this section, we investigate two case of network structures; a small scale network with 200 resources and a medium-scale network with 2048 resources.

The small-scale network structures are shown in Figures 1 to 3. In these network structures, 100 CPU resources and 100 memory resources are connected. Each switch has 10 ports connected to multi-core fibers with 4 fiber cores.

Similarly, we construct medium-scale network structures. We construct 3-level Fat-tree, 4^4 4D HyperX, and $2^4 \times 4^2$ 6D Torus as the network structures between switches. We connect 1024 CPU resources and 1024 memory resources. Each switch has 14 ports connected to multi-core fibers with 4 fiber cores.

3.3.2 Characteristics of network structure

We investigate the characteristics of network structures by calculating the metrics defined above. Figures4 and 5 shows the average number of memory resources within k hops from each computational resource. These figures show that each computational resource can access more memory resources within a shorter number of hops in the alternate arrangement. These figures also show that network structures based on HyperX enable computational resources to access more memory resources within a shorter number of hops; computational resources can access a large number of memory resources within 3 hops in the network

structure based on 2D HyperX, while only 10 and 40 memory resources can be accessed from the computational resources within 3 hops. Similarly, each computational resources can access more memory resources within a shorter number of hops in the 4D HyperX, compared with the 6D Torus and 3-level FatTree.

Figures6 and 7 show the maximum value of weighted edge betweenness centrality among the links between switches. These figure show that the alternate arrangement achieves the smaller weighted edge betweenness. This is because the number of hops between the computational and memory resources is small in the alternate arrangement. As a result, the amount of communication passing each link becomes also small.

Figures6 and 7 also show that the 2-Level/3Level Fat-Trees achieves the smallest edge betweenness centrality for large k. That is, the network structure based on Fat-Tree accommodate a large amount of communication traffic when all computational and memory resources communicate. However, considering the case that only resources close to each other communicate frequently, we focus on the case of a small k. 3D HyperX with alternate arrangement has the smallest weighted edge betweenness centrality among the small-scale network structures if k is smaller than 50. In the 3-level Fat-tree used in this investigation, each computational nodes has 14 memory resources connected to the same switch. That is, the links between switches are not used if each computational resource communicates with the 14 closest memory resources. But if $k = 15$, the edge betweenness centrality becomes large. On the other hand, 4D HyperX and 6D Torus also keeps the small edge betweenness centrality. Therefore, the network structure based on the HyperX with the alternate arrangement is suitable to accommodate more communication traffic between computational and memory resources among the small-scale network structures. Among the medium-scale network structures, in addition to the HyperX, the Torus is also a suitable network structure to accommodate more communication traffic.

Finally, Tables1 and 2 show the switch and link costs for each small-scale and medium-scale network structures, respectively. This table shows that the switch cost and link cost of 2-Level Fat-tree are higher than those of 3D Torus and 2D HyperX in the small network. But the switch and link costs of 3-Level Fat-tree are the smallest among the medium-size network structures used in this thesis. That is, the network structure with the minimum costs depends on the scale of the network; the network structure that connects a small

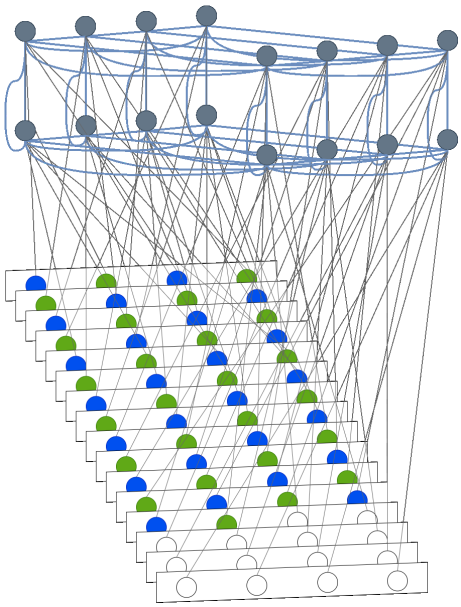
number of resources cannot always be applied to a large scale network.

Table 1: Link and switch costs for each small-scale network structure

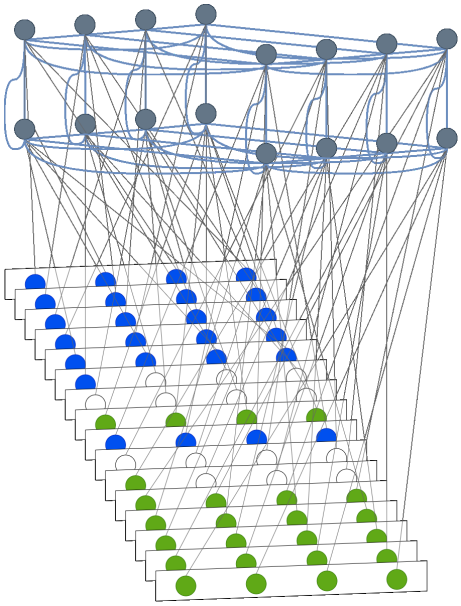
	3D Torus	2D HyperX	2-Level Fat-tree
Link cost	0.4375	0.4375	0.5000
Switch cost	0.0625	0.0625	0.0800

Table 2: Link and switch costs for each medium-scale network structure

	6D Torus	4D HyperX	3-Level Fat-tree
Link cost	0.5000	0.7500	0.6699
Switch cost	0.1250	0.1250	0.1196

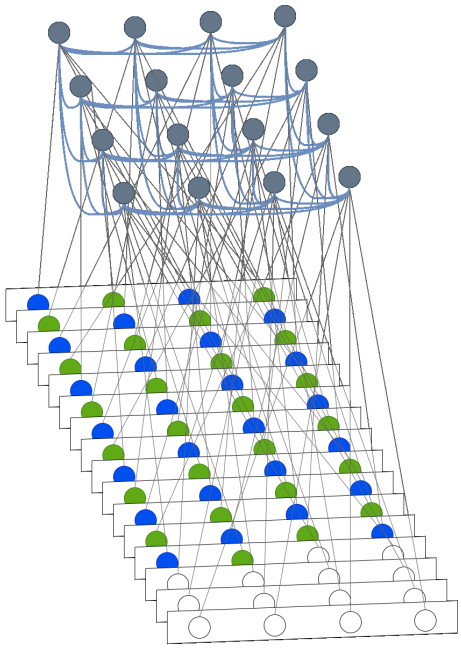


(a) Alternate arrangement

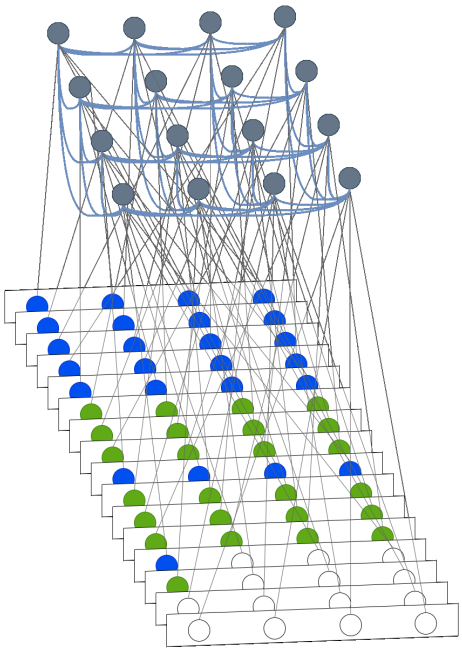


(b) Clustered arrangement

Figure 1: 3D Torus

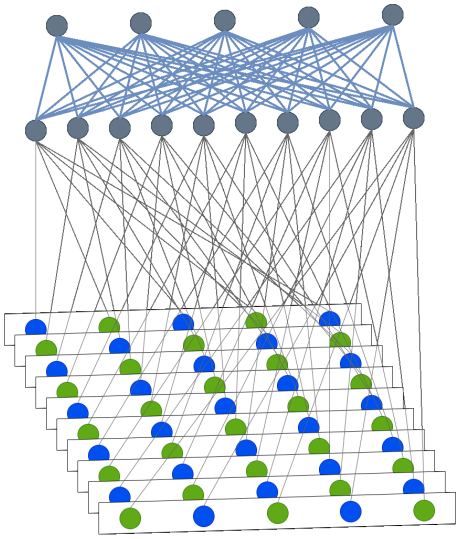


(a) Alternate arrangement

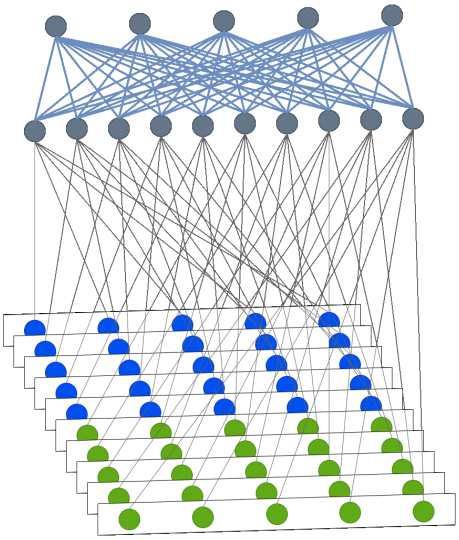


(b) Clustered arrangement

Figure 2: 2D HyperX



(a) Alternate arrangement



(b) Clustered arrangement

Figure 3: 2-Level Fat-tree

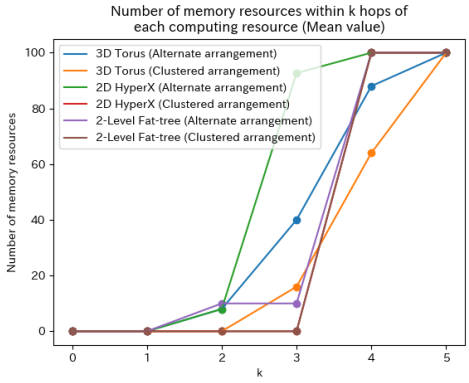


Figure 4: Number of memory resources within k hops of each computing resource for each small-scale network structure

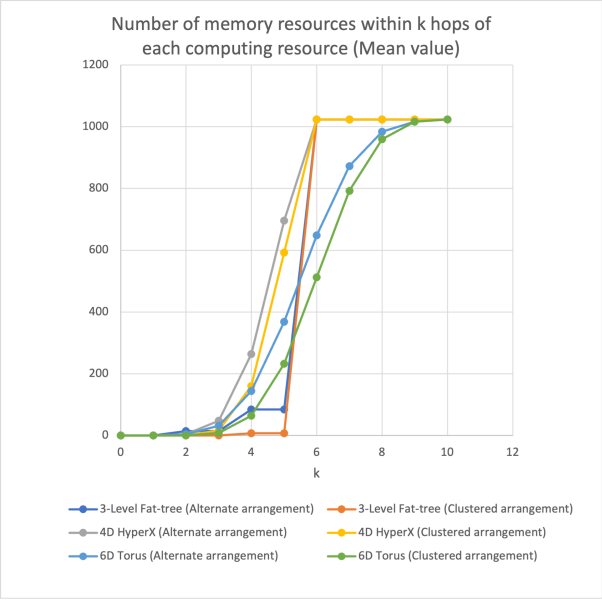


Figure 5: Number of memory resources within k hops of each computing resource for each medium-scale network structure

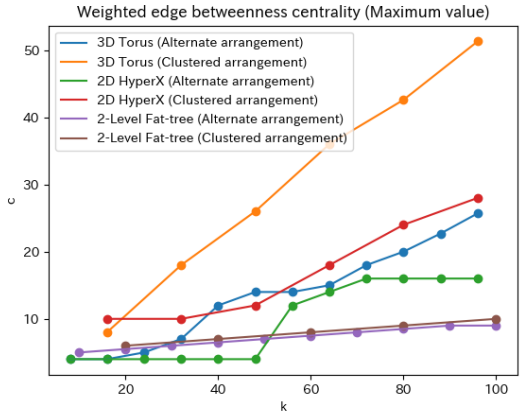


Figure 6: Weighted edge betweenness centrality for each small-scale network structure

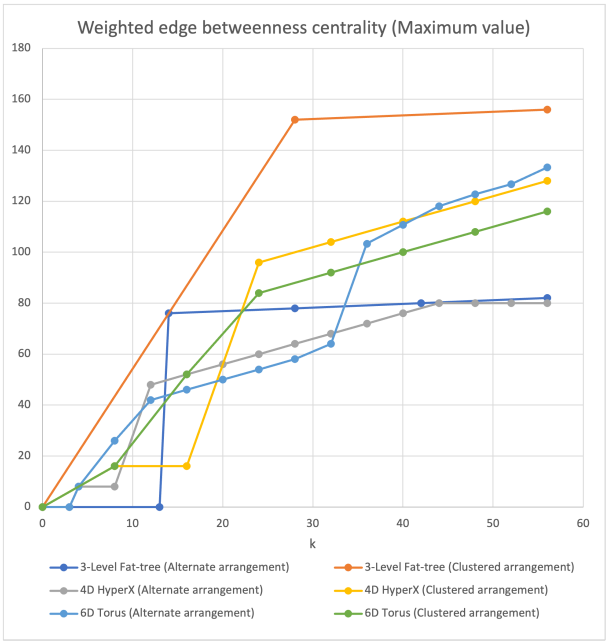


Figure 7: Weighted edge betweenness centrality for each medium-scale network structure

Table 3: Parameter settings for evaluation network

Parameter	Value
FLOPS of CPUs	13.619GFLOPS
Propagation delay	0.025mus
Processing delay of switches	3mus
Switch processing delay during cut-through	300ns
Page size	4KB
Bandwidth	50Gbps

3.3.3 Evaluation by simulation of resource allocation

In this section, we also evaluate the network structures by using the simulation of the resource allocation. In this thesis, we use the resource allocation method proposed by Ikoma et al. [7]. In this method, the resources are allocated so as to satisfy the requirement of the processing time. In this section, we generate the requests of the resources. If the resources that satisfy the requirements are found, the resources are allocated. Otherwise, the requests are blocked. By comparing the number of blocks, we investigate the number of applications that can be provided by each network structure.

Simulation settings

Resources in the network In this evaluation, we use the performance of the computational, memory and network resources shown in Table3.

Resource request In this evaluation, we generate three types of the resource requests as shown in Table4. After the requested time slot, each application ends and releases the allocated resources after the resources are allocated.

We simulate the resource allocation for 300 minutes. For the small-scale network, We generate 170 requests whose corresponding applications end 90 minutes after the allocation. For the medium-scale network, we generate 89 requests whose corresponding applications do not end before the end of simulation so as to simulate the case that many

Table 4: Parameter settings for resource request

	Request type 1	Request type 2	Request type 3
Target processing time in small-scale network	500ms	250ms	150ms
Target processing time in medium-scale network	500ms	300ms	250ms
Computing resource requirement	3	3	3
Memory resource requirement	3	3	3
Process 1			
Clock frequency	0.035	0.035	0.035
Packet arrival rate for memory writes per ms	0.00033	0.002	0.005
Packet arrival rate for memory reads per ms	0.00033	0.002	0.005
Process 2			
Clock frequency	0.054	0.054	0.054
Packet arrival rate for memory writes per ms	0	0	0
Packet arrival rate for memory reads per ms	0.00033	0.002	0.005
Process 3			
Clock frequency	2371.33	1960.36	1960.36
Packet arrival rate for memory writes per ms	1.87	1.9	1.9
Packet arrival rate for memory reads per ms	3.71	3.43	3.43
Number of page fault occurrences	67543.25	56661.29	56661.29
Number of pages communicated per page fault	5.27	4.84	4.84

applications run simultaneously. We set the probabilities that the arrived request is type 1, 2, and 3 to 0.35, 0.30, and 0.35, respectively.

Resource allocation method We used the method proposed by Ikoma et al. [7]. They modeled the time required to complete the process by the sum of the processing time in the computational resources and the time required to obtain the required information. This method allocates the resources so as to satisfy the requirement that the processing time estimated by the model is less than the time constraint. Among the candidates of the resource allocation satisfying the requirements, this method select the solution with the minimum cost. By setting the cost of important resources to a large values, this method avoids using the important resources that will be required to accommodate future requests. As a result, this method accommodates more requests.

However, the resource allocation problem to find the solution with minimum cost

Table 5: Parameter settings for resource allocation method

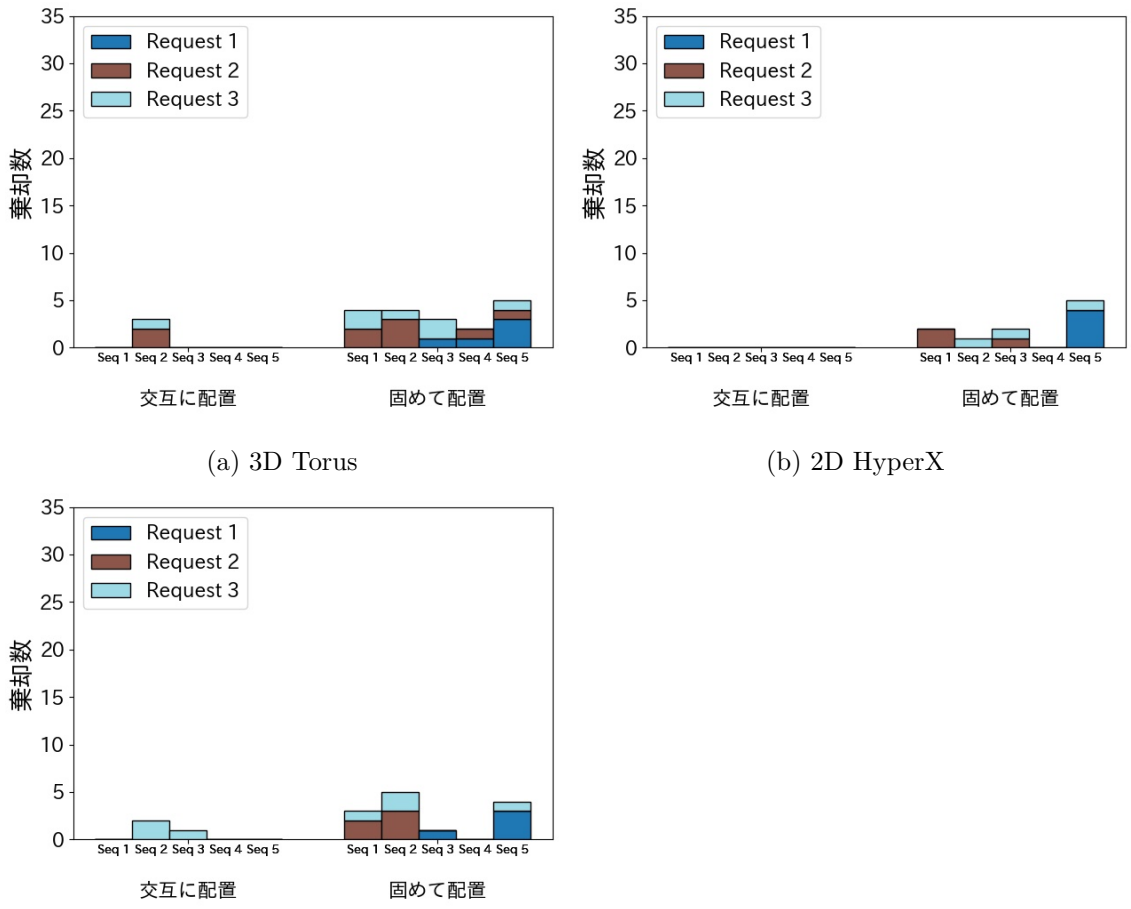
Parameter	Value
Number of agents to search for resources	20
Number of agents to search for routes	20
Number of search iterations	20
Pheromone decay rate	0.1
Pheromone enhancement rate	100
Weight of pheromone	2
Weight of cost	1
Initial value of pheromone	1000

is a nonlinear integer programming problem and NP-hard. Therefore, they solved this problem using one of the metaheuristic methods, Ant Colony Optimization (ACO). This method searches for the solution of the resource allocation by generating multiple agents. Each agent searches for resources and finds resource allocation that can finish the process within an acceptable time. Based on each agent's resource allocation, the pheromone is updated and the optimal resource is allocated. By continuing the search and update of the pheromone, the optimal solution is found.

In this thesis, we set the parameters related to ACO to the values shown in Table5.

Result Figure 8 and 9 compare the number of blocked requests. These figure show that the number of blocked request is large in the clustered arrangement. This result support the above discussion that the clustered arrangement is not suitable for the μDDC, considering our metrics. This is because the number of hops between computational and memory resources increases, which not only increases latency but also reduces the number of unused links. As a result, it becomes impossible to allocate resources to new requests.

These figure also show that the network structure based on HyperX achieves the smallest number of blocks among the network structures compared in this section. This result also supports our discussion. As discussed above, each computational resources can access the most memory resources with a small number of hops in the network structures based



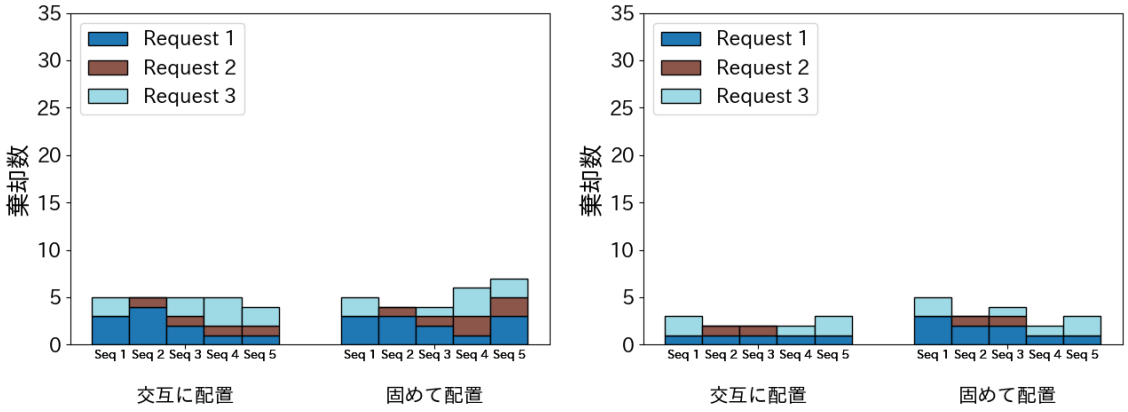
(a) 3D Torus

(b) 2D HyperX

(c) 2-Level Fat-tree

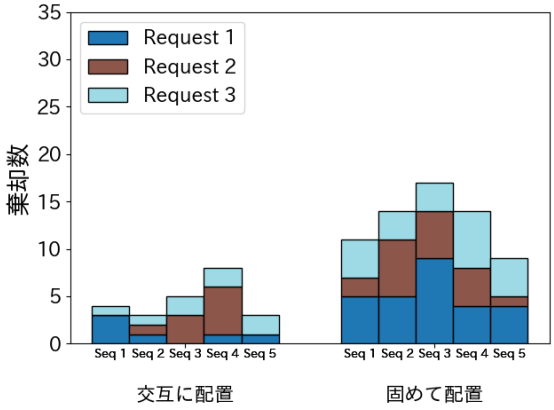
Figure 8: Number of rejections for each small-scale network structure per request type and resource requirement

on the HyperX. In addition, the weighted edge betweenness centrality is also small in the network structures based on the HyperX.



(a) 6D Torus

(b) 4D HyperX



(c) 3-Level Fat-tree

Figure 9: Number of rejections for each medium-scale network structure per request type and resource requirement

4 Disccussion and conclusion

Effective use of limited resources is required to run a wide variety of applications because resources in μDC are limited. Therefore, a resource disaggregated micro data center (μDDC) has been proposed. In a μDDC, resources such as CPU, GPU, memory, and storage are disaggregated, and connected via a network. Resources can be allocated to each application only as needed.

The network in a μDDC has a significant impact on the performance of the applications running in the μDDC. Therefore, in this thesis, we discussed a suitable network structure for μDDC. We first discussed the requirements for the network in the μDDC. Then, we introduced the metrics corresponding to the requirements for the network structures in a μDDC. We also performed simulation of the resource allocations in some network structures. The results show that (1) the number of memory resources within a small number of hops from each computational resources and (2) weighed edge betweenness centrality focusing on the communication only between near resources are important.

In this thesis, we only compare the network structures based on the structures commonly used in the high-performance computing and the data centers. But more suitable network structure may exist. Our metrics can be used to search such suitable network structures.

One approach to finding such network structures is to rewire the links of the base network structure, which is one of our future research topics. In this approach, the base network structure can be selected from the well-known network structures. According to the results in the previous section, HyperX can be used as the base network structure. Then the parameters of the base network structure are also searched based on our metrics. Finally, more suitable network structure are searched for by rewiring the links based on our metrics; if the rewiring improves the weighted edge betweenness centrality and the number of memory resources within a small number of hops from computational resources, the rewiring is adopted. By continuing the rewiring, the suitable network structure can be found.

Acknowledgments

We would like to thank Professor Masayuki Murata of the Graduate School of Information Science and Technology at Osaka University. He took time for me and gave me good advice and suggestions for my research. It has been a great help to me in my research. I would like to thank him again.

I would also like to express my sincere appreciation to Associate Professor Yuichi Ohsita of the Graduate School of Information Science and Technology, Osaka University. He advised me not only on my research but also on many other aspects as a researcher. I could not have completed this thesis without his advice.

Finally, I would like to thank all the members of the Advanced Network Architec- ture Research Laboratory at the Graduate School of Information Science and Technology, Osaka University, for creating a comfortable daily research environment and for their discussions and advice.

References

[1] S. Han, N. Egi, A. Panda, S. Ratnasamy, G. Shi, and S. Shenker, "Network support for resource disaggregation in next-generation datacenters," in *Proceedings of the Twelfth ACM Workshop on Hot Topics in Networks*, 2013, pp. 1–7.

[2] H. Yuan, A. Saljoghei, A. Peters, and G. Zervas, "Disaggregated optical data center in a box network using parallel ocs topologies," in *Optical Fiber Communication Conference*. Optical Society of America, 2018, pp. W1C–2.

[3] Q. Cheng, M. Bahadori, M. Glick, S. Rumley, and K. Bergman, "Recent advances in optical technologies for data centers: a review," *Optica*, vol. 5, no. 11, pp. 1354–1370, 2018.

[4] P. X. Gao, A. Narayan, S. Karandikar, J. Carreira, S. Han, R. Agarwal, S. Ratnasamy, and S. Shenker, "Network requirements for resource disaggregation." in *OSDI*, vol. 16, 2016, pp. 249–264.

[5] G. Zervas, H. Yuan, A. Saljoghei, Q. Chen, and V. Mishra, "Optically disaggregated data centers with minimal remote memory latency: technologies, architectures, and resource allocation," *Journal of Optical Communications and Networking*, vol. 10, no. 2, pp. A270–A285, 2018.

[6] A. Ikoma, Y. Ohsita, and M. Murata, "Impact of remote memory and network performance on execution performance of disaggregated micro data centers," in *Proceedings of 2021 International Conference on Emerging Technologies for Communications*, 2021, pp. C2–2.

[7] Y. O. Akishige Ikoma and M. Murata, "Disaggregated Micro Data Center: Resource Allocation Considering Impact of Network on Performance," in *Proceedings of IEEE 20th Consumer Communications Networking Conference (CCNC)*, Jan. 2023.

[8] J. H. Ahn, N. Binkert, A. Davis, M. McLaren, and R. S. Schreiber, "Hyperx: topology, routing, and packaging of efficient large-scale networks," in *Proceedings of the Conference on High Performance Computing Networking, Storage and Analysis*, 2009, pp. 1–11.

[9] M. Al-Fares, A. Loukissas, and A. Vahdat, "A scalable, commodity data center network architecture," *ACM SIGCOMM computer communication review*, vol. 38, no. 4, pp. 63–74, 2008.

[10] J. Kim, W. J. Dally, and D. Abts, "Flattened butterfly: a cost-efficient topology for high-radix networks," in *Proceedings of the 34th annual international symposium on Computer architecture*, 2007, pp. 126–137.

[11] V. Mishra, J. L. Benjamin, and G. Zervas, "Monet: heterogeneous memory over optical network for large-scale data center resource disaggregation," *Journal of Optical Communications and Networking*, vol. 13, no. 5, pp. 126–139, 2021.