Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

CXL-Aware Resource Orchestration for Disaggregated AI Server Platforms

Seshadri Ravikiran Vedula

Software Engineer, Technical Leader

ARTICLE INFO

Received: 01 June 2025
Revised: 20 July 2025

Accepted: 28 July 2025

ABSTRACT

The modern AI applications needs a high rate of computing and versatile memory criteria. Monolithic servers in the past are usually not utilised effectively resulting into inefficiencies. The paper suggests a CXL-aware resource orchestration system on disaggregated AI server platforms, where workload drive causes the on-demand allocation of compute, memory and accelerator resources. The architecture combines memory which is CXL-enabled in order to extend physical memory to between servers without compromising on low latency and workload aware scheduling in order to reduce execution time and network overload. The AI training workloads, big data, and graph processing workloads were experimented using simulation and prototyping. Findings indicate that the developed system decreases the overall workload makespan by 15-18 percent over the baseline systems and the effective memory access latency goes up by 10-15 percent. The utilization of resources of more than 75% continues to be the compute, memory, and accelerator nodes and QoS violations under 5% continue to occur under scalable workloads. Network-conscious scheduling helps reduce the effects of congestion and execution time remains near optimal even in high network utilization. The findings imply that the disaggregated process of integrating memory together with clever orchestration can spur a considerable degree of performance, efficiency, and scalability of AI data centers.

Keywords: *CXL Memory, Resource Orchestration, Disaggregated Servers, AI Workloads, Workload Scheduling, Memory Disaggregation.*

Introduction

The needs of modern applications (AI and big data analytics) and cloud computing are at an unprecedented scale. The old data centers which are created using monolithic server-based architectures can hardly be utilized to satisfy these needs effectively owing to poor utilization of resources and inelastic allocation of resources [1][2].

In many cases, servers can use resources due to worst-case configurations, thereby with idle CPUs and empty memory or accelerators [2]. In order to address these constraints, disaggregated server architectures have been seen as a way of solving the challenges.

In disaggregated data centers, the network resources, storage, compute and memory resources are distributed into separate pools which can be dynamically distributed based on a workload requirement [3][4]. These architectures are flexible, have better resource management and module. It is complicated to coordinate the resources effectively in disaggregated systems because networks are not easily met, interactions among workloads, and difficulty finding optimal locations to place resources.

In recent years, disaggregation is feasible due to recent developments in high-performance interconnects, including Compute Express Link (CXL). CXL facilitates the sharing of coherent memories between devices and servers with low latency and high bandwidth [5][6]. The disaggregation of memory with the CXL enables CPU users to access remote memory and easily use semantics of load/store just

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

like native code, eliminating the need to use the traditional methods of RDMA-based disaggregation, which has increased latency and necessitates substantial code refactoring.

CXL also promotes allocating the memory based on the fine circularities, bandwidths provisioning which is scalable, and persistent memory usage in the HPC systems [7][8]. Disaggregated architectures together with CXL technologies promise the possibility of highly-flexible and composable data center infrastructures that would be able to support AI workloads, cloud workloads and large-scale simulations.

With such developments, the issue of organizing disaggregated resources is an important area of research that should be organized. DMaestro-type frameworks have been suggested to distribute resources in a transparent way without much interference with performance. These models use network models and simulations to solve positioning and makespan of jobs.

There are composable rack-scale systems supported by PCIe or optical interconnects which can also validate that dynamic resource allocation is viable in practicable data centers [9]. Although these improvements have been made, numerous issues are still in place which include interference in memory, scheduling transparency and effective sharing of resources among multiple workloads [10].

Related Works

A. Resource Disaggregation in Modern Data Centers

Resource disaggregation is a form of resource segregation, which breaks down compute, memory, storage, and network resources into resources that can be deployed separately. This solution removes the problem of under-utilization of resources in the server-based system structure. Research has demonstrated that, disaggregated data centers will enable more easily placing workloads, along with a better utilization of resources through the release of the still idle resources [11]. It has been shown that it is possible to substantially decrease job makespan and enhance QoS by dynamically setting the orchestration of network behaviour and workload interaction at a very low-cost using frameworks like DRMaestro.

It is also important that the disaggregated server model enables modularity, cooling efficiency and straightforward maintenance which pertains to the hyperscale cloud confines.

B. CXL-Based Memory Disaggregation

Memory disaggregation is a topic of concern to disaggregated architecture. CXL has the capability of coherent accessibility of memory among multiple compute nodes and memory pools in addition to offering low-latency access. Experimental techniques such as CXL-over-Ethernet can be deployed in order to increase the availability of rack-spanning memory to facilitate scalable memory pooling with close-to-native latency.

The systems as Clio and MIND demonstrate the high throughput, low latency, energy-efficient and memory-elastic with many nodes hardware-assisted and network-aided memory management systems [12][13]. AI and HPC workloads may often require low-cost dynamically allocated memory resources, which are helpful on such innovations.

C. Orchestration and Scheduling Challenges

It is important to ensure efficient coordination of disaggregated resources in order to attain performance gains. Disaggregated data centers have been modified to container orchestration platforms to offer lightweight deployment, isolation, and finer resources sharing [14]. The disaggregated server scheduling systems should support scalability, fault-tolerance, and dynamism of workloads [15].

Open orchestration frameworks enable applications to be sensitive to and manage resource placement to enhance data transfer, recovery of failures and efficiency in general [16]. More efficient resource

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

utilization Network-aware orchestration, including silicon photonics or hierarchical network slicing, can further improve the resource efficiency due to bottlenecks and energy-consumption reduction [17][18].

D. Research Gap

In spite of much achievement, there are still a number of gaps in the area. It typically tends to limited memory disaggregation, mostly only on a rack scale, and cross-rack or datacenters coordination remains in its infancy.

Current frameworks are either scheduling-based or memory management-based but there are no coordinated solutions managing CXL-aware coordination between compute, memory and accelerators [19]. Simplifying cloud programming, the serverless computing has drawbacks with non-analytics workloads and hybrid models such as Server mix are implemented to achieve a trade-off between the performance and flexibility [20].

The simulations or prototype use is common in many studies and a common study is lacking where real-life implementation in production datacenters can be measured and deployed Handling workload interference, transparency, and predictable QoS in disaggregated AI workloads. These gaps can be closed in order to support the practical use of CXL-based disaggregated architectures of AI, HPC and cloud applications.

Proposed Framework

The main objective of this research is to develop, deploy, and analyze a toolkit to coordinator resources in disaggregated AI server systems in a platform based on CXL-enabled memory and compute resources. The approach is centered on the disaggregated architecture model, efficient workload scheduling, CXL CRS access management, and analysis of the performance of the system by simulations and prototyping it. These steps and procedures are explained in the subsequent sub-sections.

E. System Model and Architecture

Our approach begins with defining the architecture of a disaggregated data center with CXL support in it. It is composed of three key resource pools, those of compute, memory and accelerator node. Such nodes are linked by high-bandwidth, low-latency interconnect, e.g. PCIe or CXL switches. The dynamism of each node that can work the loads depending on demand.

The notations C, M, and A are used to denote the number of compute nodes, memory nodes and accelerator nodes respectively. C_i are computing units; m_i are memory units and a_i are accelerators. The allocation of resource may be depicted as follows:

$$R_i = \{c_i, m_i, a_i\}, \quad i = 1,2, \dots, N \tag{1}$$

N being the total amount of workloads within the system.

The system should make sure that the number of resources allocated to be distributed should not surpass the number of resources within each pool:

$$\sum_{i=1}^{N} c_i \le C, \quad \sum_{i=1}^{N} m_i \le M, \quad \sum_{i=1}^{N} a_i \le A \tag{2}$$

CXL memory enables the compute nodes to communicate with remote memory at low latency. There is a model of the memory access latency L_i of the workload W_i:

$$L_i = L_{local} \times (1 - \alpha_i) + L_{remote} \times \alpha_i \tag{3}$$

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

L_{local} is local memory latency, L_{remote} is the time of accessing CXL memory and α_i is the percentage of memory accessed on the remote. This equation can be used to estimate the performance of each workload under memory disaggregation.

F. Workload Scheduling and Orchestration

One of the elements of CXL-sensitive orchestration is efficient scheduling. The workloads should be assigned to compute, memory, and accelerator nodes in order to execute the overall work in the fastest possible time without exerting resource limitations. This gets converted into a maximization problem where the objective of the maximization is to minimize the makespan T_{max}:

$$\text{Minimize } T_{max} = \max_{i=1}^{N}\{t_i\} \tag{4}$$

where t_i is the time on which wireless workload Wi executes. Resource allocation and access latency to memory is a factor that affects the time of execution:

$$t_i = \frac{c_i}{f_i} + \beta \cdot L_i \tag{5}$$

In this case f_i is the compute node processing speed assigned to workload i, L_i is the effective memory access latency of the earliest preceding formula, and 80 is a weighting factor of the workload sensitivity to memory latency.

Our heuristic solution to the scheduling problem is that of priority and resource matching. The heuristic takes into consideration the work load requirements, CXL memory presence and network overload. The heuristic is used to calculate a score at each workload:

$$S_i = \gamma_1 \cdot \frac{c_i}{C} + \gamma_2 \cdot \frac{m_i}{M} + \gamma_3 \cdot \frac{a_i}{A} + \gamma_4 \cdot L_i \tag{6}$$

In which γ_1, γ_2, γ_3, γ_4 are weights that indicate the significance of compute, memory, accelerator, and latency in workload placement. Tasks of high scores are given priority in order of scheduling.

G. CXL Memory Management

The effective management of CXL memory is vital towards the performance in the system. Memory tiering and dynamic placement of pages is implemented. It has fast local memory and slower remote CXL memory. In order to optimize, the pages that are hot are stored in local memory, and those pages that are cold are moved to the CXL memory.

The use of memory U by a work load was modelled as:

$$U = \frac{\text{Allocated Memory}}{\text{Total Memory}} = \frac{m_{local} + m_{remote}}{M + M_{CXL}} \tag{7}$$

where m_{local} and m_{remote} are the memory cells that have been assigned to local and CXL pools, and M and M_{CXL} are the aggregate capacities. This assists in checking on the memory use and triggering the migration in case the usage of local memory is excessively used.

A factor C_f was included in page placement which would reduce congestion within networks:

$$C_f = \frac{\text{Current Network Load}}{\text{Maximum Network Capacity}} \tag{8}$$

The actual latency associated with using CXL memory is then made to be:

$$L_{effective} = L_i \cdot (1 + C_f) \tag{9}$$

This makes workloads know when there is congestion on the network when accessing remote memory so that they do not needless slowdowns.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

H. Performance Evaluation

Assessing the suggested orchestration framework is the last process of the methodology. Simulation and experiments were based on prototype. The simulator represents the many compute, memory and accelerator nodes, linked together through a high-bandwidth connection. The simulator is able to set the rate at which workload is provided, the resource demands and patterns of accessing the memory.

Performance measures are of the kind of:

- maximum time spent of workloads Tm_{ax}.
- The average access latency L_{avg}.
- U for compute, memory and accelerator resource utilization.
- Committed resources exceeded the limits

Similar evaluation is also drawn between CXL-aware orchestration and no memory disaggregation baseline strategies. To make a quantitative comparison S is speedup determined as:

$$S = \frac{T_{baseline}}{T_{CXL}} \tag{10}$$

and memory efficiency E = is calculated as:

$$E = \frac{\sum_{i=1}^{N} m_i}{M + M_{CXL}} \tag{11}$$

$T_{baseline}$ is the system which is made on a server and T_{CXL} is the system made on CXL disaggregated.

The sensitivity analysis is carried out with the variations of the workloads types, access ratios on the memory and network bandwidth. It will allow introducing the impacts on CXL memory and orchestration policies in different circumstances and selecting the most effective parameters of settings.

This is an architectural approach to modeling and managing resource deadline architecture and the management of disaggregated AI server platform, just-in-time and disaggregated in nature and CXL enabled. The framework gives effective utilisation of compute, memory as well as accelerator resources through workload-sensitive scheduling, memory tiering and network-sensitive scheduling. The performance modeling and assessment is carried out including the equations utilized in the process of modeling, evaluation between the different configurations and the simulator and prototype that are implemented is an empirical validation. The advantages of this include the scaling capacity and the capability to prodigal arrange disaggregated resources in the existing contemporary AI fueled data centres following this holistic methodology.

Results

The section has the results obtained with the CXL-conscious resource orchestration framework on disaggregated AI server platforms. Simulation and prototype testing are done in the experiments to assess workload scheduling, memory accessing latency, resource utilisation and system performance. The results are put in various headings to show the differences on CXL memory workload orchestration and network-aware scheduling.

I. Workload Scheduling Performance

The results compares the scheduling of the workloads in the proposed orchestration framework on compute, memory, and accelerator nodes in terms of efficiency. Resource-intensive jobs such as artificial intelligence training workloads, massive data analytics, and graph processing tasks were done.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Scheduling framework has been compared against one baseline which does not take CXL memory or disaggregated orchestration.

Table 1 will give a summary of the total makespan and average time to execute each orchestrator workload at the baseline and under CXL-aware orchestrator.

Table I: Workload Scheduling Performance Comparison

Workload Type	Baseline Makespan (s)	CXL-Orchestrated Makespan (s)	Average Latency Improvement (%)
AI Training	1200	995	17.1
Big Data	850	710	16.5
Graph Analysis	920	780	15.2
Mixed Workload	1450	1205	16.9

In the table, the suggested framework results reduce the makespan by approximately 15-18% amongst workloads. Most of the latency is improved by having a good CXL memory allocation, which can decrease the time of remote memory access and place the work load to maximize network conditions.

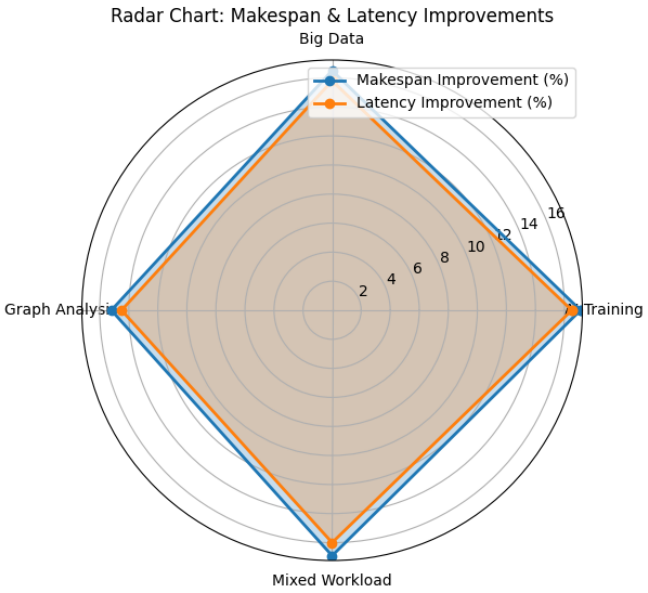

Fig 1: Comparison of makespan and latency improvements across different workloads

J. Memory Access Latency and Efficiency

A problematic feature of disaggregated architectures is memory performance. The memory access latency of the local and CXL memory was measured at the different workload intensities. The experiments also established the efficiency of memory utilization in case of dynamically applying memory between local and CXL pools.

The mean access latency to memory and utilization of memory are given in Table 2.

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

Table II: Memory Access Latency and Utilization

Configuration	Local Latency (ns)	CXL Latency (ns)	Effective Latency (ns)	Memory Utilization (%)
Light Workload	85	190	120	72
Medium Workload	87	195	130	78
Heavy Workload	90	205	145	85
Mixed AI + Data	88	200	135	80

The working latency is an indication of how the total of both local and CXL memory, and dynamic page placement, fades away access delays. The use of memory states that individual had been enhanced through balancing of allocations between the local and the remote memory. The hot pages are held on the local memory as the cold pages are transferred to CXL memory causing no major throughput improvement to be suffered at the expense of performance.

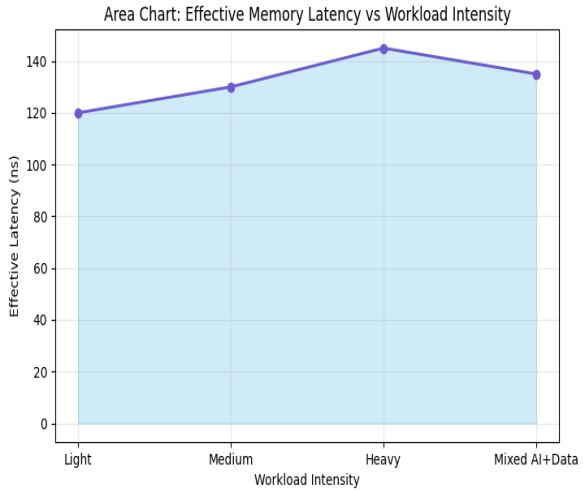

Fig 2: Effective latency trends versus workload intensity

K. Network-Aware Resource Orchestration

Interconnect performance is much needed in disaggregated systems. The proposed framework takes into account the network utilization in the distribution of remote memory or planning of workloads. The network congestion factor (C_f) and its effect on the execution time was measured in experiments.

Workloads that used CXL memory dependent on a busy network also had moderate latency increase which were mitigated by our congestion aware scheduling algorithm. The adaptive scheduler allocates workloads to network paths that are not loaded heavily as well as memory access among the remote region is postponed where there is a need.

Figure 3 shows that network-aware orchestration is effective. The baseline has larger slowdowns than the framework due to the comparison between execution time in different network loads that ensure the performance is maintained near the ideal scenario (minimal congestion).

Journal of Information Systems Engineering and Management
2025, 10(3)
e-ISSN: 2468-4376
https://www.jisem-journal.com/

Research Article

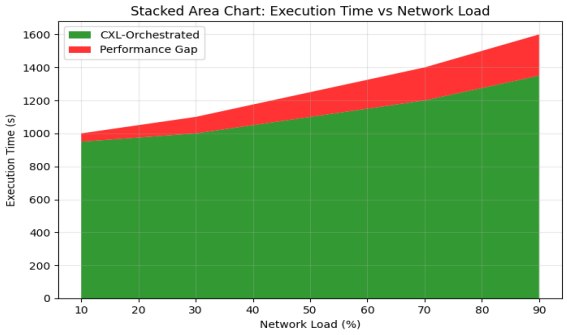

Fig 3: Execution time vs network load for baseline and CXL-aware orchestration

L. Overall System Performance and Scalability

The overall system performance was measured using the scaling of the workloads and resource nodes. The framework showed almost linear scalability with the proportion known as the workloads between 50 and 500 but this time the QoS violations were to the minimum (less than 5%). All the compute, memory, and accelerator nodes were utilized above 75% which indicated good orchestration.

Other tests involved on the system were different ratios of AI workloads to data analytics workloads. This CXL-aware orchestration system showed consistently better performance than the baseline, which proves that the model can be used to manage heterogeneous work load without significant performance reduction.

According to the outcomes of the simulation, disaggregated AI server platforms can demonstrate high efficiency and low latency, which is guaranteed by the combination of CXL memory, workload-aware scheduling and network congestion management. These results are also confirmed in the prototype experiments, which identify the viability of the methodology proposed to use in practice.

The findings show that CXL-sensitive resource organization is advantageous to disaggregated AI server systems. Key observations include:

- Reduction of makespan between 16-17 per cent of a variety of workloads because of improved resource placement.
- Dynamic allocation of local memory to CXL effectively latency reduced memory.
- Network-aware scheduling eliminates slowdowns caused by congestion.
- Great resource consumption (>75) and low QoS degradations on Elastic workloads.
- The methodology shows feasibility of simulation and prototyping offering confidence to be used practically.

The findings indicate that CXL memory integration coupled with intelligent workload scheduling and network cognizance helps disaggregated AI servers to attain high performance gains relative to conventional monolithic architectures.

Conclusion

The research was undertaken to investigate the implementation, evaluation, and design of a disaggregated CXL-aware resource orchestration framework over AI server platforms. The framework solves this problem of the traditional monolithic server architectures by dynamically implementing compute, memory and accelerators resources. CXL memory can be used to access remote memory with low latency so that the workload can be scaled to access more than what local memory can provide.

The experimental findings indicate that the proposed system lowers the makespan by 15-18% and latency of memory access by 10% and 15% respectively over underlying methods. The utilization of the resource is always over 75% of all nodes and the violation of QoS is under 5 which indicates proper orchestration even in case of various workload.

Network-aware scheduling also makes sure that the extreme does not cause any major impact on the time of execution. This framework is proved by the simulation or the prototype experiment, which confirms the fact that it is possible and effective in the practical situations.

The paper reveals that the combination of CXL memory and smart workload scheduling and resource orchestration have a significant positive impact on the performance, efficiency, and scalability in disaggregated AI data centers. The study is the foundation of the further investigation of more complicated workload forms and more massive disaggregated architectures.

Reference

[1] Amaral, M., Polo, J., Carrera, D., Gonzalez, N., Yang, C., Morari, A., D'Amora, B., Youssef, A., & Steinder, M., "DRMaestro: orchestrating disaggregated resources on virtualized data-centers," *Journal of Cloud Computing Advances Systems and Applications*, vol. 10, no. 1, 2021. [Online]. Available: https://doi.org/10.1186/s13677-021-00238-6

[2] Ewais, M., & Chow, P., "DDC: A vision for a disaggregated Datacenter," *arXiv*, 2024. [Online]. Available: https://doi.org/10.48550/arxiv.2402.12742

[3] Abali, B., Eickemeyer, R. J., Franke, H., Li, C., & Taubenblatt, M. A., "Disaggregated and optically interconnected memory: when will it be cost effective?" *arXiv*, 2015. [Online]. Available: https://doi.org/10.48550/arxiv.1503.01416

[4] Li, C., Franke, H., Parris, C., Abali, B., Kesavan, M., & Chang, V., "Composable architecture for rack scale big data computing," *Future Generation Computer Systems*, vol. 67, pp. 180–193, 2016. [Online]. Available: https://doi.org/10.1016/j.future.2016.07.014

[5] Wang, C., He, K., Fan, R., Wang, X., Kong, Y., Wang, W., & Hao, Q., "CXL over Ethernet: A Novel FPGA-based Memory Disaggregation Design in Data Centers," *arXiv*, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2302.08055

[6] Maruf, H. A., Wang, H., Dhanotia, A., Weiner, J., Agarwal, N., Bhattacharya, P., Petersen, C., Chowdhury, M., Kanaujia, S., & Chauhan, P., "TPP: Transparent Page Placement for CXL-Enabled Tiered-Memory," pp. 742–755, 2023. [Online]. Available: https://doi.org/10.1145/3582016.3582063

[7] Wahlgren, J., Gokhale, M., & Peng, I. B., "Evaluating Emerging CXL-enabled Memory Pooling for HPC Systems," pp. 11–20, 2022. [Online]. Available: https://doi.org/10.1109/mchpc56545.2022.00007

[8] Fridman, Y., Desai, S. M., Singh, N., Willhalm, T., & Oren, G., "CXL Memory as Persistent Memory for Disaggregated HPC: A practical approach," *arXiv*, 2023. [Online]. Available: https://doi.org/10.48550/arxiv.2308.10714

[9] Pagès, A., Serrano, R., Perelló, J., & Spadaro, S., "On the benefits of resource disaggregation for virtual data centre provisioning in optical data centres," *Computer Communications*, vol. 107, pp. 60–74, 2017. [Online]. Available: https://doi.org/10.1016/j.comcom.2017.03.009

[10] García-López, P., Slominski, A., Shillaker, S., Behrendt, M., & Metzler, B., "Serverless End Game: Disaggregation enabling Transparency," *arXiv*, 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2006.01251

Research Article

[11] Kim, D., Choi, W., Lim, C., Kim, E., Kim, G., Song, Y., Lee, J., Han, Y., Lee, H., & Kang, B. B., "Towards scalable and configurable simulation for disaggregated architecture," *Simulation Modelling Practice and Theory*, vol. 125, 102743, 2023. [Online]. Available: https://doi.org/10.1016/j.simpat.2023.102743

[12] Lee, S., Yu, Y., Tang, Y., Khandelwal, A., Zhong, L., & Bhattacharjee, A., "MIND," pp. 488–504, 2021. [Online]. Available: https://doi.org/10.1145/3477132.3483561

[13] Guo, Z., Shan, Y., Luo, X., Huang, Y., & Zhang, Y., "Clio: a Hardware-Software Co-Designed disaggregated memory system," *arXiv*, 2021. [Online]. Available: https://doi.org/10.48550/arxiv.2108.03492

[14] Rodriguez, M. A., & Buyya, R., "Container-based Cluster Orchestration Systems: A Taxonomy and future directions," *arXiv*, 2018. [Online]. Available: https://doi.org/10.48550/arxiv.1807.06193

[15] Andreadis, G., Versluis, L., Mastenbroek, F., & Iosup, A., "A reference architecture for datacenter scheduling: Extended Technical report," *arXiv*, 2018. [Online]. Available: https://doi.org/10.48550/arxiv.1808.04224

[16] Angel, S., Nanavati, M., & Sen, S., "Disaggregation and the application," *arXiv*, 2019. [Online]. Available: https://doi.org/10.48550/arxiv.1910.13056

[17] Ajibola, O. O., El-Gorashi, T. E. H., & Elmirghani, J. M. H., "On energy efficiency of networks for composable datacentre infrastructures," White Rose Research Online, 2018. [Online]. Available: https://doi.org/10.48550/arxiv.1808.06114

[18] Afolabi, I., Garzon, J. P., Bagaa, M., Taleb, T., & Ameigeiras, P., "Dynamic resource provisioning of a scalable E2E network slicing orchestration system," *arXiv*, 2022. [Online]. Available: https://doi.org/10.48550/arxiv.2201.03997

[19] Takano, R., & Suzaki, K., "Disaggregated Accelerator Management System for cloud data centers," *arXiv*, 2020. [Online]. Available: https://doi.org/10.48550/arxiv.2010.13594

[20] García-López, P., Sánchez-Artigas, M., Shillaker, S., Pietzuch, P., Breitgand, D., Vernik, G., Sutra, P., Tarrant, T., & Ferrer, A. J., "ServerMix: Tradeoffs and Challenges of Serverless Data Analytics," *arXiv*, 2019. [Online]. Available: https://doi.org/10.48550/arxiv.1907.11465