Disaggregation Patterns for Secure AI Systems

Mohamed Islam Ghamri
Orange Innovation
mohamed.ghamri@orange.com

Marc Lacoste
Orange Innovation
marc.lacoste@orange.com

Divi De Lacour
Orange Innovation
divi1.delacour@orange.com

***Abstract*—Disaggregation is a growing trend in large-scale artificial intelligence (AI) systems to overcome hardware and software resource limitations and improve performance while preserving security and privacy. This paper takes a closer look at different dimensions of the concept, in AI, security and hardware. We identify two key design patterns that may be combined to build optimized disaggregated AI architectures and discuss benefits and limitations for AI and security. Using a large language model use case, we also highlight some key trade-offs between performance, resource allocation and security for different disaggregation strategies in hardware and in software.**

I. Disaggregation in Computing

Researchers and developers have consistently faced the challenge of resource limitations in hardware and in software. Extensively explored, e.g., in software engineering and in networking, *disaggregation* is emerging as a key solution for large-scale artificial intelligence systems. This concept may be defined as separating a system into smaller elements or *components*. Foreseen benefits include optimization of resources, security and performance. Disaggregation may be applied to *AI*, *hardware* and *security* (see Figure 1).

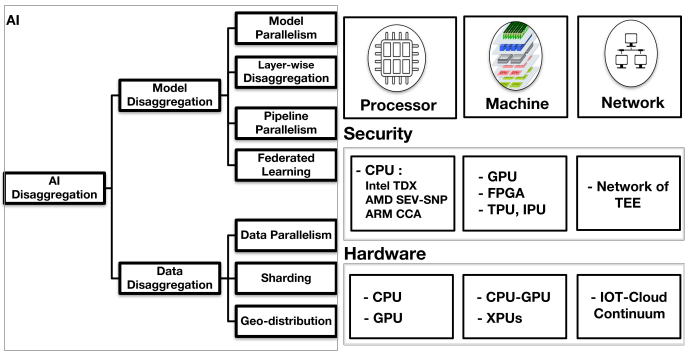

Fig. 1. Overview of disaggregation: AI, hardware and security

AI Disaggregation. Software disaggregation distributes workloads across multiple computing units. This approach enhances computational capabilities to handle complex AI models and large datasets [1].

1) Model disaggregation distributes computations performed in neural networks to improve efficiency and scalability. This set of techniques applies to inference, training and fine-tuning. For instance, *model parallelism* distributes model parameters across multiple devices – enabling simultaneous processing of different parts of models too large to fit into single-device memory. *Layer-wise disaggregation* adopts different parallelization configurations for different layers of a neural network, e.g., for convolutional layers and fully-connected layers. *Pipeline parallelism* divides a model into segments processed sequentially across hardware units, reducing significantly training time. Finally, *Federated Learning* [2] collaboratively trains a model across decentralized devices and is expected to improve data privacy.

2) Data disaggregation is another approach that distributes data across storage and computations for model training and deployment, with similar efficiency and scalability benefits.

Data parallelism replicates data across processors or machines, each copy handling a subset of the data. Training may scale out with faster convergence and management of larger datasets. *Sharding* partitions a large dataset into smaller elements that can be processed in parallel – improving performance, e.g., for database management systems. Finally, *geo-distribution* policies specify spatial data placement across data centers to reduce latency, increase fault tolerance and ensure compliance with data sovereignty regulations.

Hardware Disaggregation. Disaggregation may occur at various hardware levels, from single accelerators to groups of accelerators (XPUs) [3] up to high-performance computing (HPC) and cloud infrastructures. Such hierarchical disaggregation enables more scalable and tailored allocation of resources.

Within the processor (multiple cores, execution environments), computing tasks may be run concurrently, optimizing processor capabilities. *On a single machine*, workloads may also be distributed across different processing units (e.g., CPUs, GPUs, other accelerators) to handle complex learning tasks. Computations may finally be expanded *between multiple machines* (in a data center, geographically dispersed) to process large-scale datasets and train large models.

Confidential Computing/AI. Security and privacy are key properties to guarantee as AI systems are increasingly distributed across multiple platforms. *At application-level*, techniques such as secure multi-party computation (SMPC) and homomorphic encryption enable private collaborative computation and encrypted data operations respectively. *At middleware level*, secure Kubernetes frameworks such as CNCF Confidential Containers (CoCo) or Constellation preserve data integrity for distributed AI workloads. *At hardware level*, trusted execution environments (TEE) [4] enhance AI security, integrating TEE secure execution with GPU acceleration.

II. AI Disaggregation Patterns

We identify two main patterns for disaggregation: *horizontal disaggregation (HD)* and *vertical disaggregation (VD)* shown in Figure 2. Those patterns capture different strategies to organize and optimize processes and resources in AI systems, both at software and hardware levels. HD and VD may be used to scale out and scale up respectively.

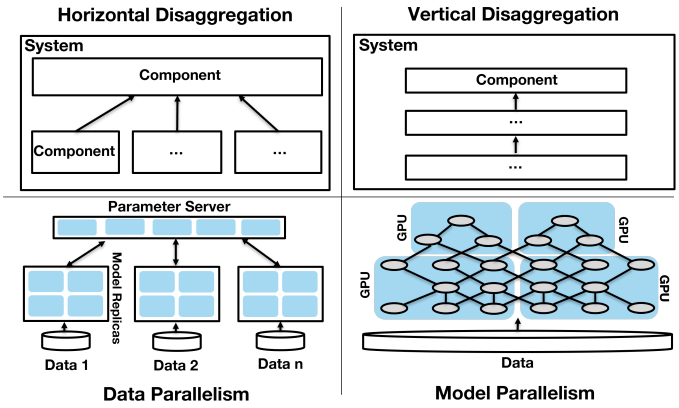

Fig. 2. Disaggregation patterns and application to data/model parallelism

Horizontal Disaggregation. In HD, tasks are distributed across multiple components, and run concurrently or in parallel, cooperating towards a common goal. HD is found in neural networks and in multi-core CPU architectures or hardware accelerators such as GPUs.

The main benefit of HD is scalability for distributed computations, reducing latency or increasing throughput. Component segmentation and distribution of responsibilities together with malicious behavior detection in aggregators can limit the impact of security breaches. Challenges include robust synchronization, as component coordination increases complexity. The attack surface may be enlarged as multiple components are interconnected.

Vertical Disaggregation. In VD, tasks are processed sequentially across different layers. Each layer is specialized in a particular function that must be completed before passing on to the next. VD is notably found in layer-by-layer propagation algorithms (forward and backward) for neural network architectures.

Benefits include layer-level optimization and upgrade of components. Isolation between layers is straightforward as the placement of protection mechanisms may be directly derived from the system architecture. The challenge is performance as each layer has to wait for the previous one for task completion.

VD vs. HD? HD and VD are usually used together. For instance, in large language models (LLM), HD is used for data preprocessing on CPU cores, and HD/VD for training acceleration with GPUs. In federated learning, VD is used for propagation through neural network layers, and HD for model aggregation.

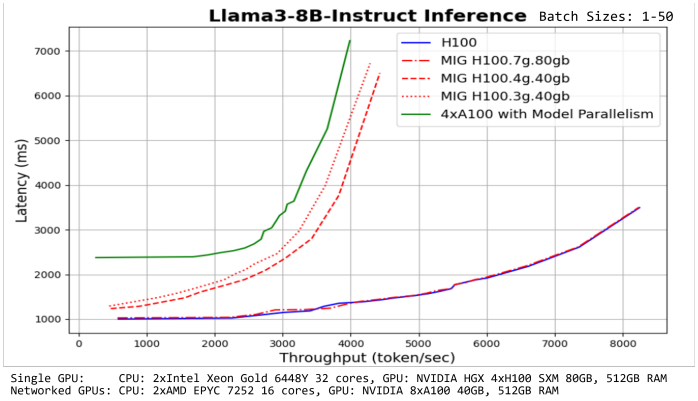

Fig. 3. Throughput w.r.t. latency for single GPU, MIG GPU slicing and networked-GPUs hardware disaggregation configurations

III. Initial Experiments and Next Steps

Hardware Disaggregation: LLMs. We developed a use case to explore the impact of AI disaggregation in hardware, focusing on LLM inference. We assess hardware performance (e.g., throughput, latency) and influence of batch size to reach optimal, secure and efficient deployment. The platform runs containerized LLM workloads using secure Kubernetes scheduling and supports several levels of disaggregation: 1) a single powerful GPU, 2) GPU partitioned into slices using NVIDIA Multi-Instance GPU (MIG) and 3) networked GPUs.

Preliminary scalability results are shown in Figure 3. The single GPU provides the best performance. Disaggregating the GPU using MIG provides flexibility to run multiple AI workloads concurrently but with performance degradation, which increases as GPU slices get smaller. Pushing hardware disaggregation further, networked GPUs over the cloud continuum using model parallelism yield even lower performance due to network communication. Increasing batch size illustrates HD performance benefits on throughput, but increasing latency. Regarding vertical scaling, the use of newer technology (H100 vs. A100 for GPUs) increases performance, in terms of latency and throughput. Such results could help finding the right disaggregation configuration depending on application requirements (e.g., latency, bandwidth, locality).

Next Steps. We plan to confirm such findings on software disaggregation in federated learning, training several AI architectures locally and on cloud platforms – exploring further security and performance dimensions.

Acknowledgment. This research was supported by the CRYPTECS project funded by French ANR and German BMBF agencies (grants ANR-20-CYAL-0006 and 16KIS1441 respectively).

References

[1] T. Ben-Nun and T. Hoefler, "Demystifying Parallel and Distributed Deep Learning: An In-depth Concurrency Analysis," *ACM Comput. Surv.*, vol. 52, no. 4, 2019.

[2] A. A. Messaoud et al., "Shielding Federated Learning Systems against Inference Attacks with ARM TrustZone," in *Middleware 2022*.

[3] K. Vaswani et al., "Confidential Computing within an AI Accelerator," in *USENIX ATC 2023*.

[4] F. Tramèr and D. Boneh, "Slalom: Fast, Verifiable and Private Execution of Neural Networks in Trusted Hardware," in *ICLR 2019*.