airai VALID 2020 October 18 – 22, 2020

Adaptive Data-Center for AI Machine Learning Acceleration

www.huawei.com

Xinli Gu

FUTUREWEI Technologies

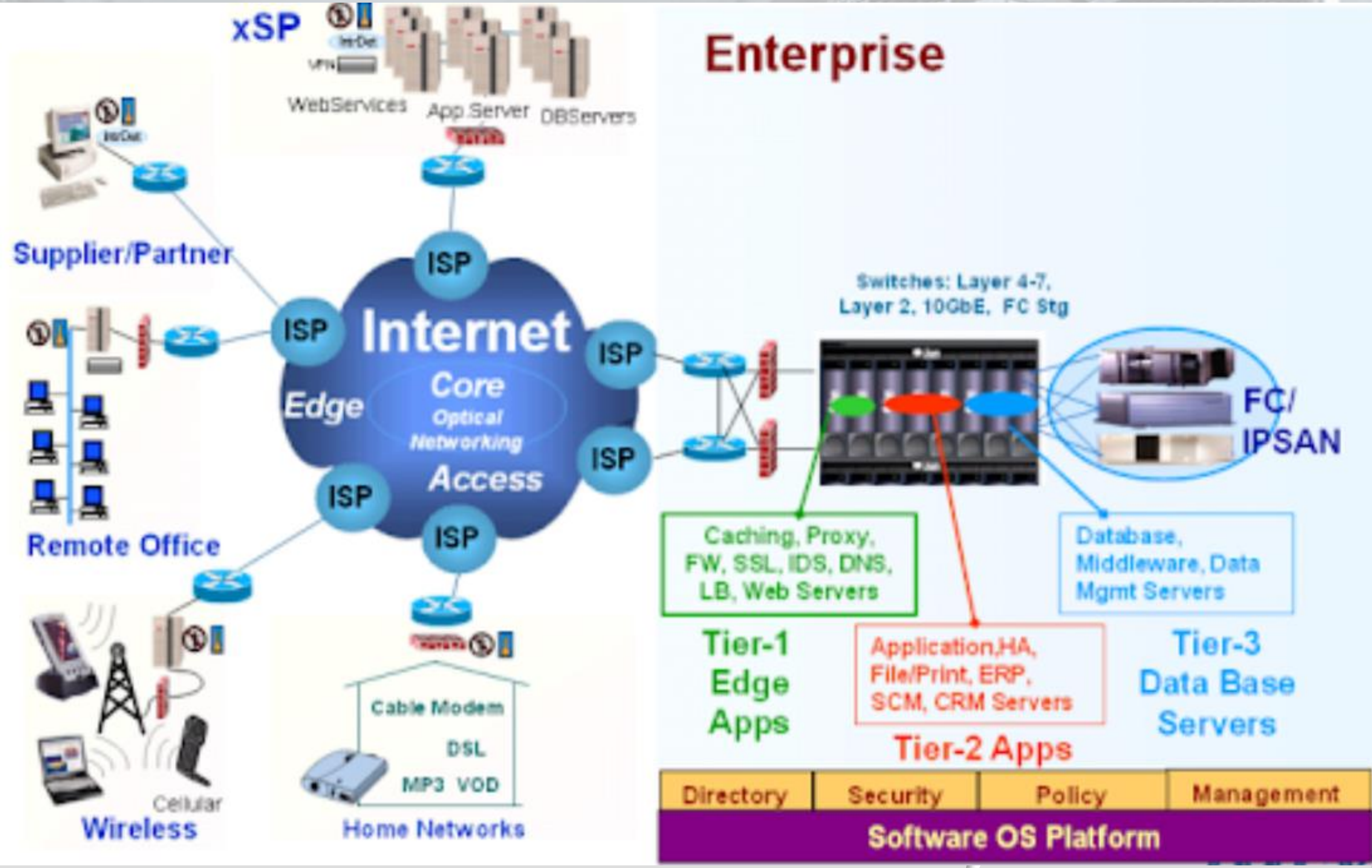
FUTUREWEI TECHNOLOGIES CO., LTD.

Outline

- **Introduction**
  - ✓ Data center (DC) applications
  - ✓ Market and development trends
- **Data Center**
  - ✓ Basic DC configuration and challenges
  - ✓ Configurable DC: Software-Defined Data Center (SDDC)
  - ✓ Adaptability using ML
  - ✓ Hardware/software codesign
  - ✓ Extended EDA for optimization
- **OCP deployment**
- **Summary**

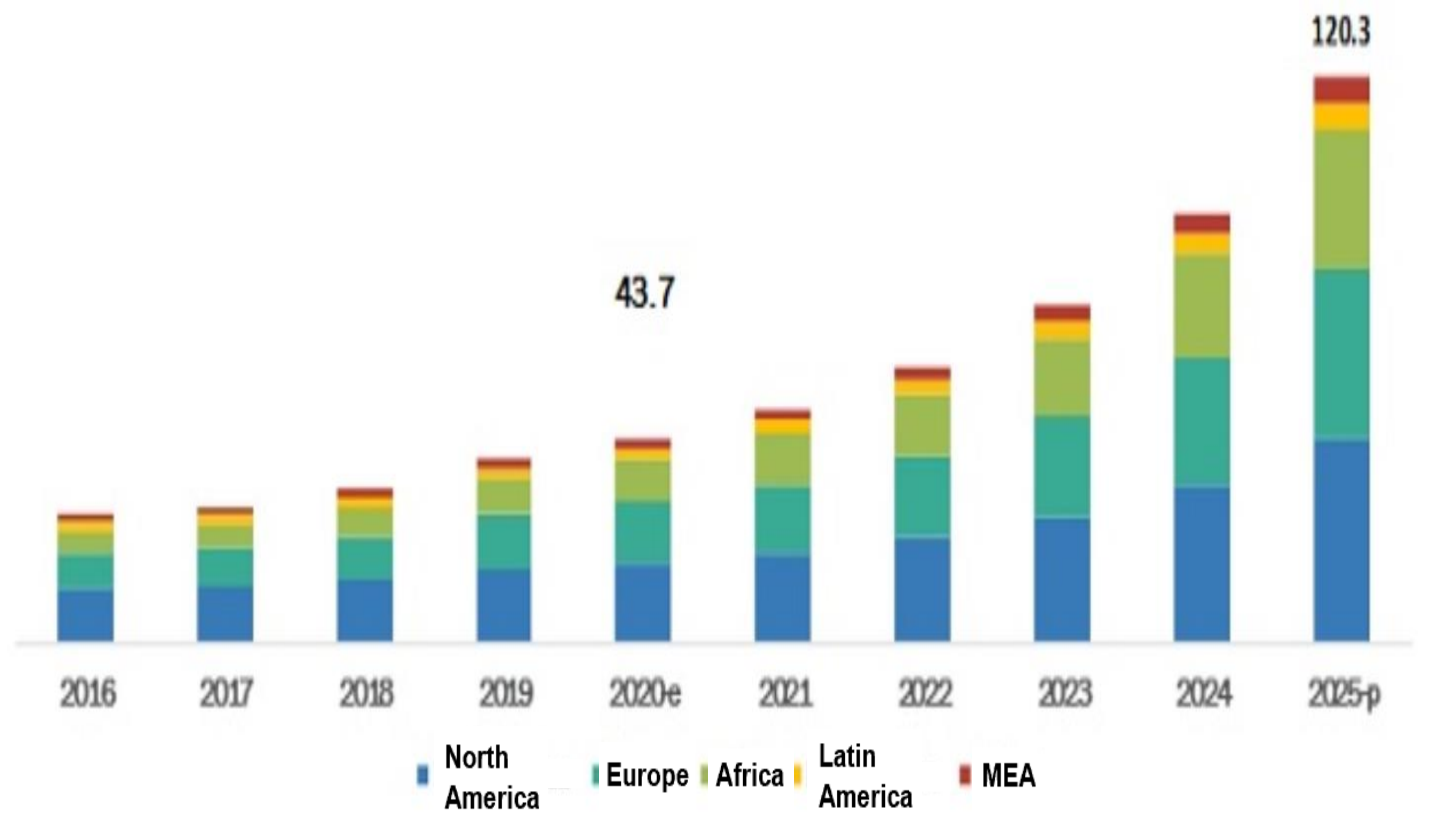
Introduction – DC Applications

- Data center provides
  - ✓ Computation
  - ✓ Storage
  - ✓ Network
  - ✓ Security
- Clouds services
  - ✓ SaaS
  - ✓ PaaS
  - ✓ Edge /Orchestra /AI
- Improve infrastructure usage and IT OPEX
  - ✓ Expected 50% improvement

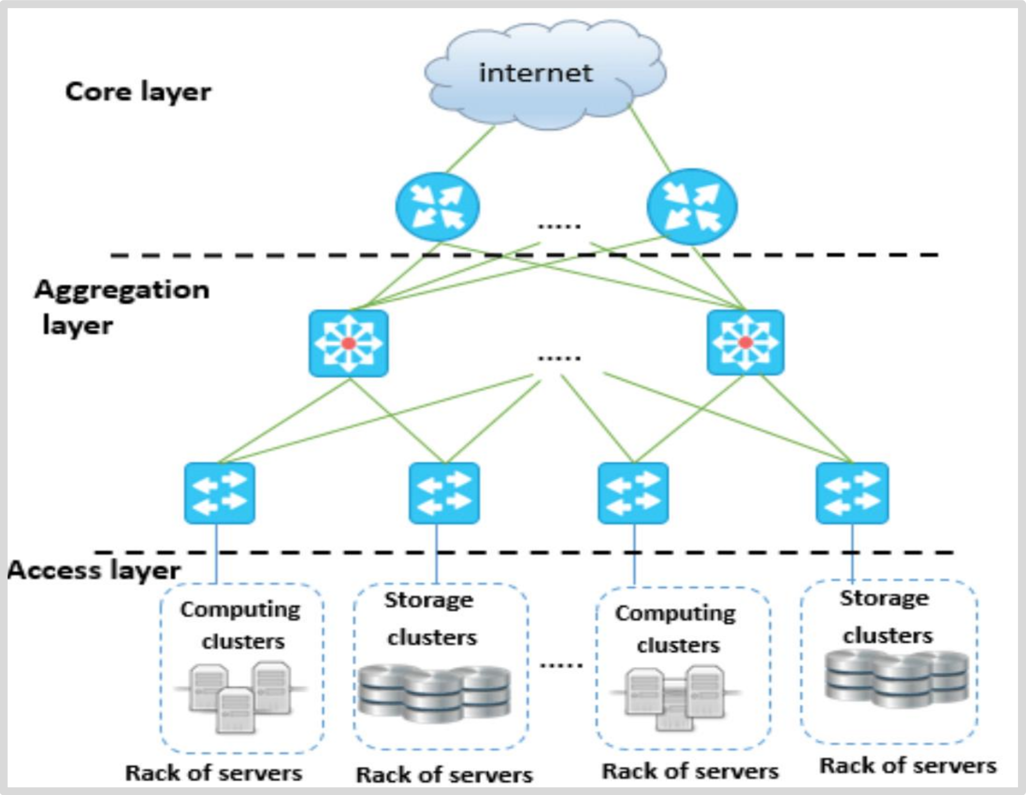


Introduction – Market

- Software-Defined Data Center (SDDC) market size is projected to grow
  - ✓ **$43.7** billion in 2020
  - ✓ **$120.3** billion by 2025
- At a Compound Annual Growth Rate (CAGR) of 22.4%
- 30 % DC failing to prepare for AI will no longer be operationally or economically viable by 2020



Introduction – DC Architectures



- Core layer: high-speed packet switching backplane going in and out of the data center
- Aggregation layer
  - ✓ Service module integration
  - ✓ Layer 2 domain definitions, spanning tree processing
  - ✓ Server-to-server multi-tier traffic flows through the aggregation layer
- Access layer: servers

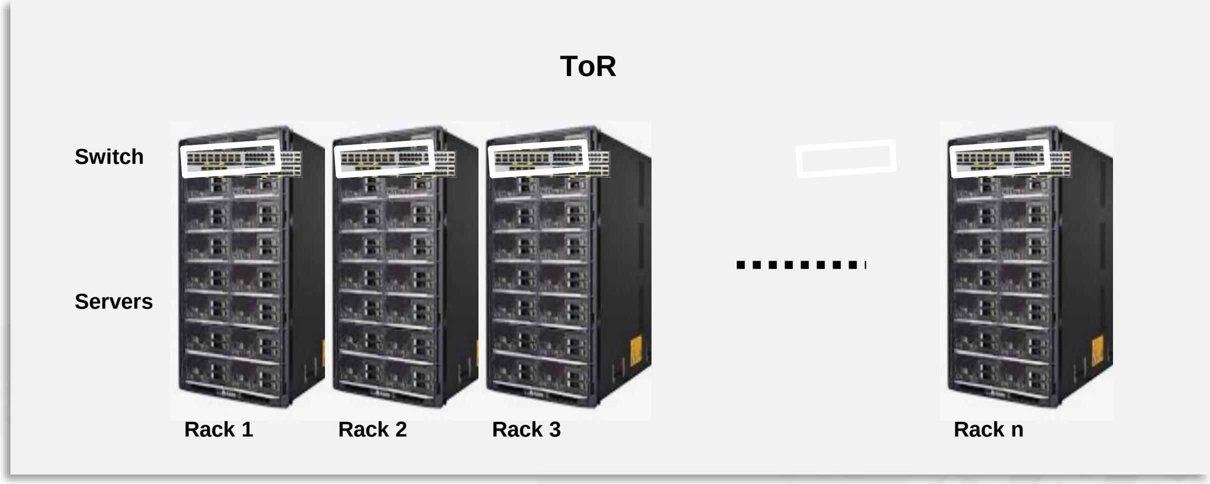
Introduction – DC Architecture Examples

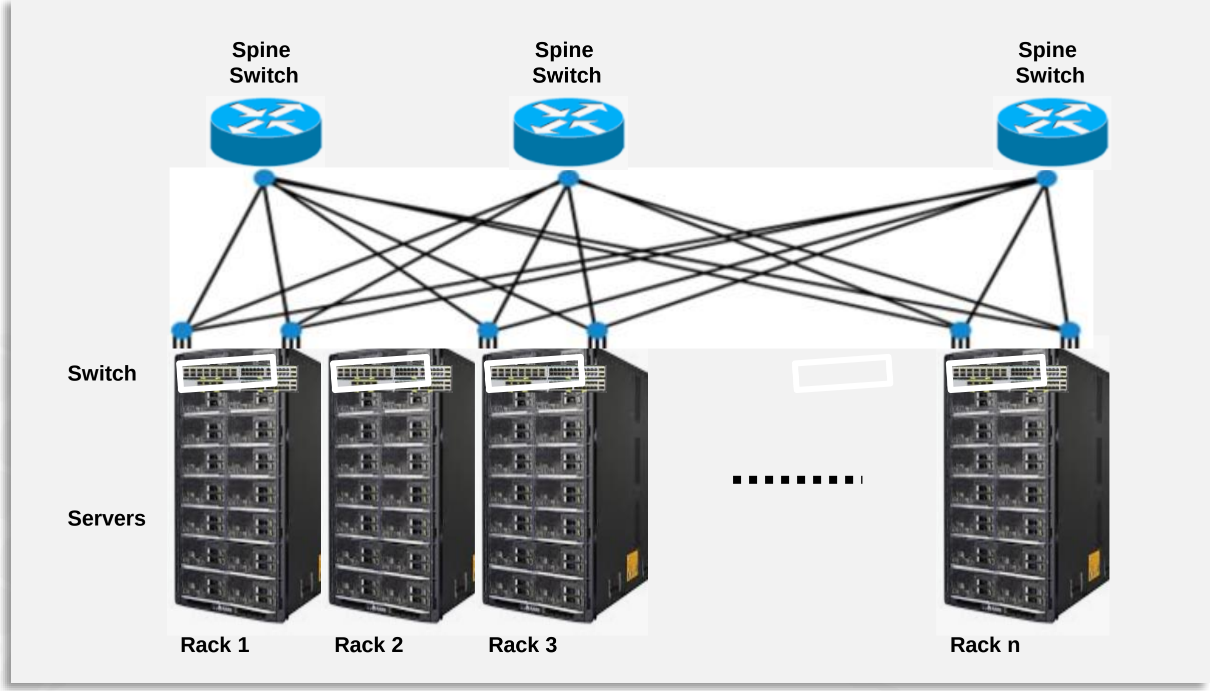
Switch-to-server connections

- Top of rack (**ToR**): one switch for each rack. Servers within the rack are connected to the switch via copper cable. All switches for the racks are connected to ToR switches, spine switches.

Features

- ✓ Copper stays "In Rack", lower cabling costs
- ✓ Modular, flexible "per rack" architecture, and higher speeds
- ✓ High capital and maintenance costs. The distributed architecture of ToR requires more physical switches
- ✓ Under utilized switches cost unnecessary power



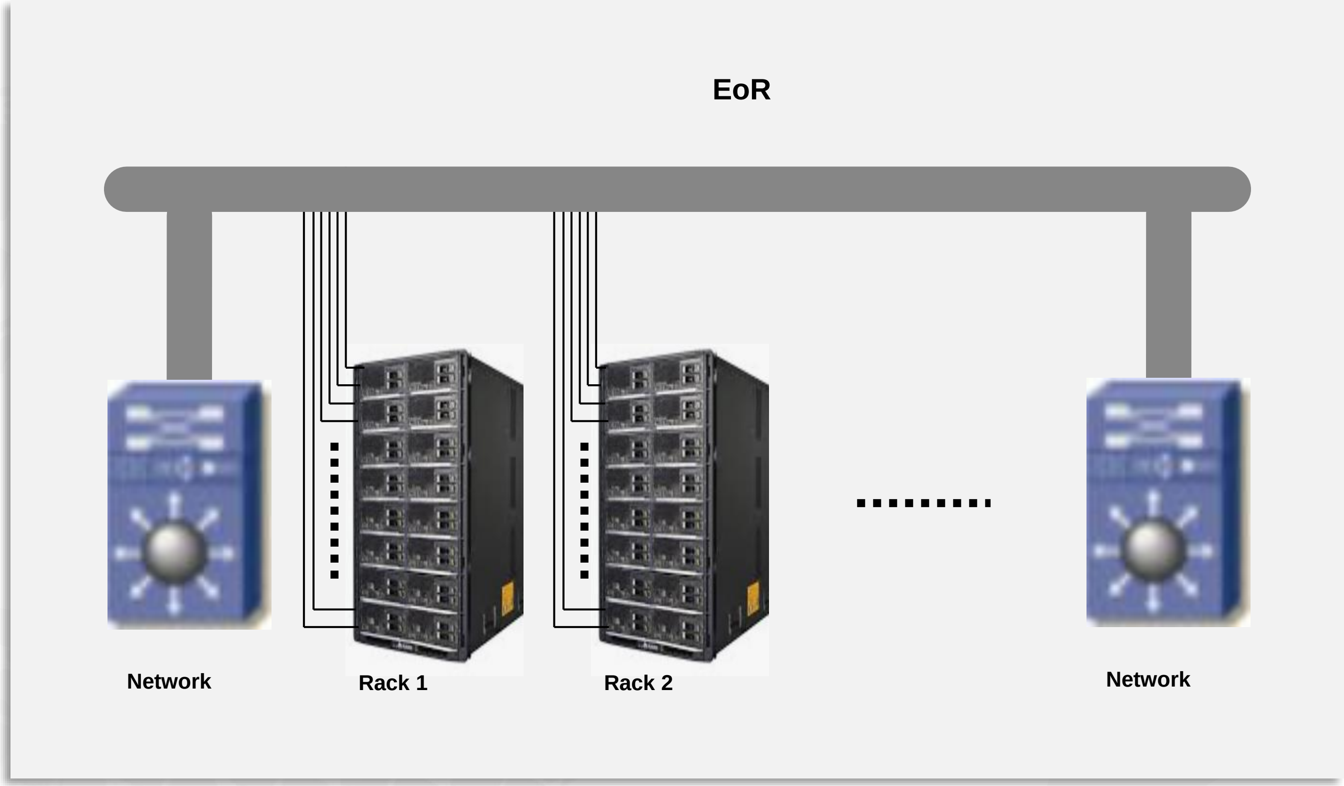


Introduction – DC Architecture Examples

- End-of-Row / Middle of Row (**EoR** / **MoR**): a dedicated networking rack at either end of a row of servers for the purpose of providing network connectivity to the servers. Within that row, both end can have a networking rack for reliability redundancy purpose

Features

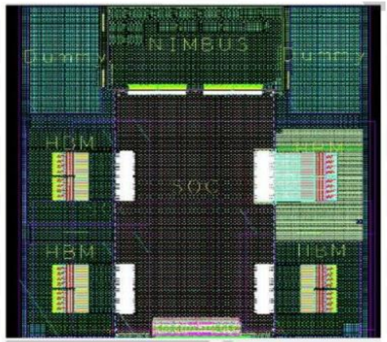
- ✓ Allows collapsing of the access and aggregation layers into a single tier of high-density chassis switches. It reduces the number of switches
- ✓ Provides improved performance by reducing the level compared to multi-tier approach
- ✓ Each server rack would have a twisted pair copper cabling routed through overhead cable trays to the switch rack

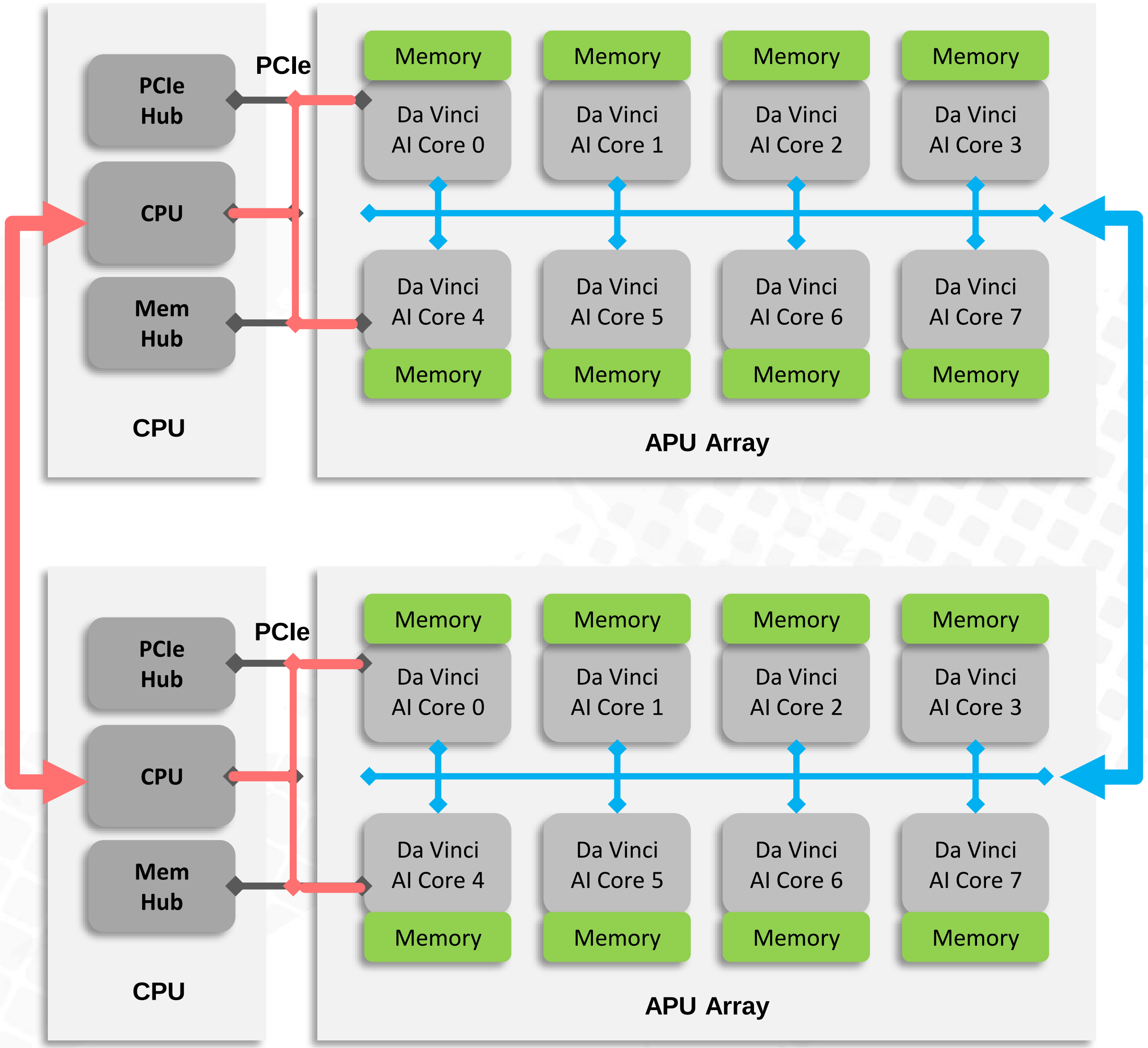


Introduction – AI Server/Core Configuration Example

Within the server, the main components are APU/GPU, memory and CPU. There are many possible configurations

- Communicate via CPU
  - ✓ Easy to synchronize
  - ✓ Low efficiency
- Communicate bypass CPU
  - ✓ High communication efficiency, especially for pipelined computations
  - ✓ Need synchronized computation output and input



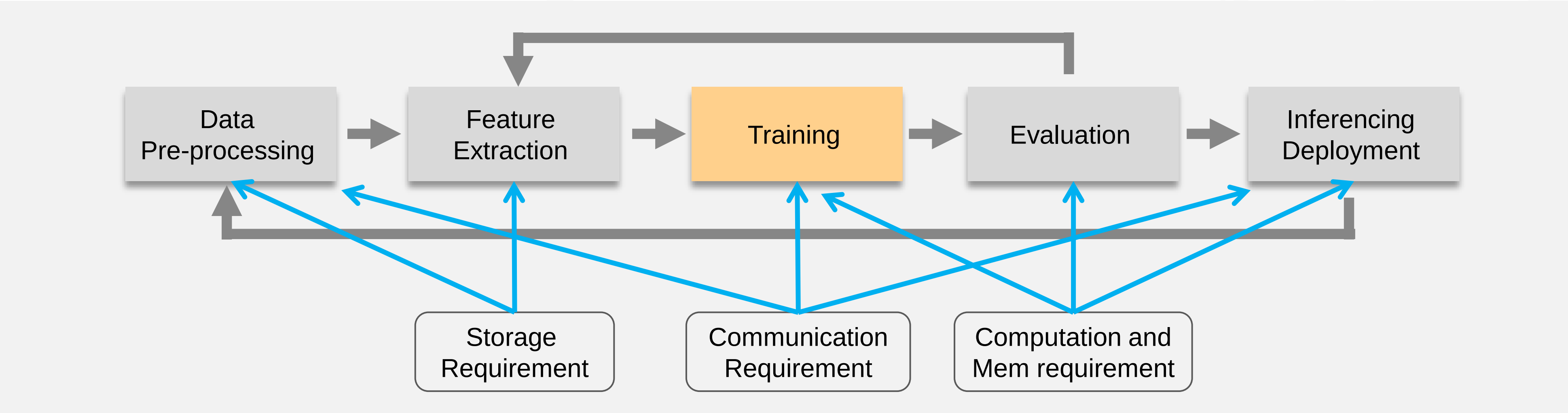


Challenges

- Exponential growth of large and complex data due to digitalization. In addition, enterprises increasingly using various software, such as ERP, CRM, and SCM create a huge amount of data related to customers, operations, suppliers, and other stakeholders. Securely storing crucial business information with flexible DC is the key to success
- Lack of universally accepted virtualization standards and different vendor hardware and cloud solutions cause
  - ✓ Integration complexity which requires skill and knowledge
  - ✓ Interoperability and efficiency is not optimized
- DC/AI core and storage configuration are not flexible enough to support heavy computation tasks, such as
  - ✓ Computer vision for image classification, object detection, and video understanding
  - ✓ Ranking and recommendation, such as news feed and search
  - ✓ Language processing for translation, speech recognition, content understanding, etc.

Challenges

- Need to support typical AI machine learning process



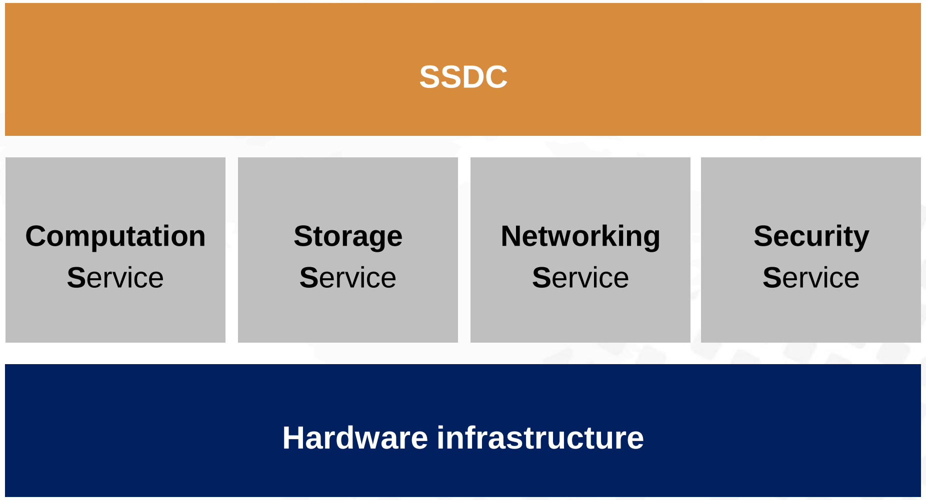
- Different operation tasks require different memory and storage configurations. Machine learning is an intensive matrix multiplication task
- Huge operation efforts for recognition of systems for
  - ✓ High capacity, high bandwidth memory
  - ✓ Unstructured accesses benefit from caches
  - ✓ Larger on-chip memory for flexibility of compiler

Strategy – Overview

- **Adaptive DC to support the secured and scalable data storage, software services and computation efficiency**
  - ✓ Flexible DC resource configuration based on applications for optimized workload management with greater agility, speed, and security
- **Automated and efficient model training and optimization without hassles associated with integration and deployment maintenance support**
  - ✓ Machine learning requires huge data set and heavy computation for model training and inferencing deployment
  - ✓ Data storage, recovery and cybersecurity along with managing large volumes needs complicated and time-consuming process. Need to better support various cloud strategies, scalability across heterogenous clouds
- **Hardware/software co-design and EDA to scale the software with programmable building hardware**
  - ✓ Concurrency and control feature, especially for many cores
  - ✓ Computation feature that supports scalar and SIMD (Single Instruction, Multiple Data)
  - ✓ Data reuse with software-controlled SRAM
  - ✓ Latency hiding such as hardware for prefetcher, etc.

SSDC (Software-Defined Data Center)

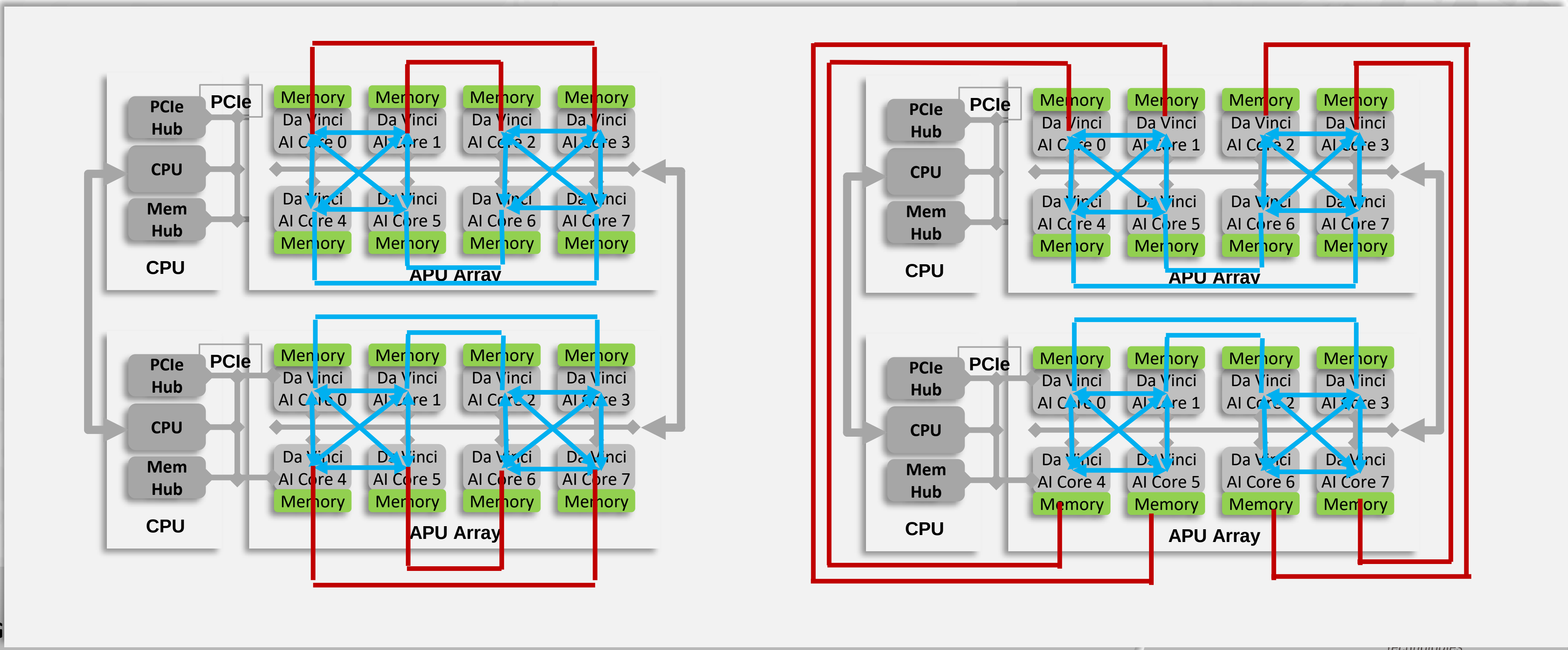
- **SDDC is a software-based data storage facility where all the resources are combined to provide the best service**
  - ✓ Core CPU/APU/GPU for computation
  - ✓ Storage for data
  - ✓ Networking for communication
  - ✓ Security



- **SSDC can be planned at hierarchical level**
  - ✓ Data center level which includes servers, storage and networking
  - ✓ Server level which includes CPU/APU, memory and data flow interconnection

SSDC – Flexible Data Flow and Communication Examples

- **Flexible PCIe interconnect topology**
  - ✓ AI core to CPU
  - ✓ AI core to AI core direct
  - ✓ AI core to AI core via CPU
  - ✓ Group of AI core to Group of AI core direct
  - ✓ Group AI core to Group of AI core via CPUs
- **Examples**



SSDC – Standards

The SDDC solution should follow these standards

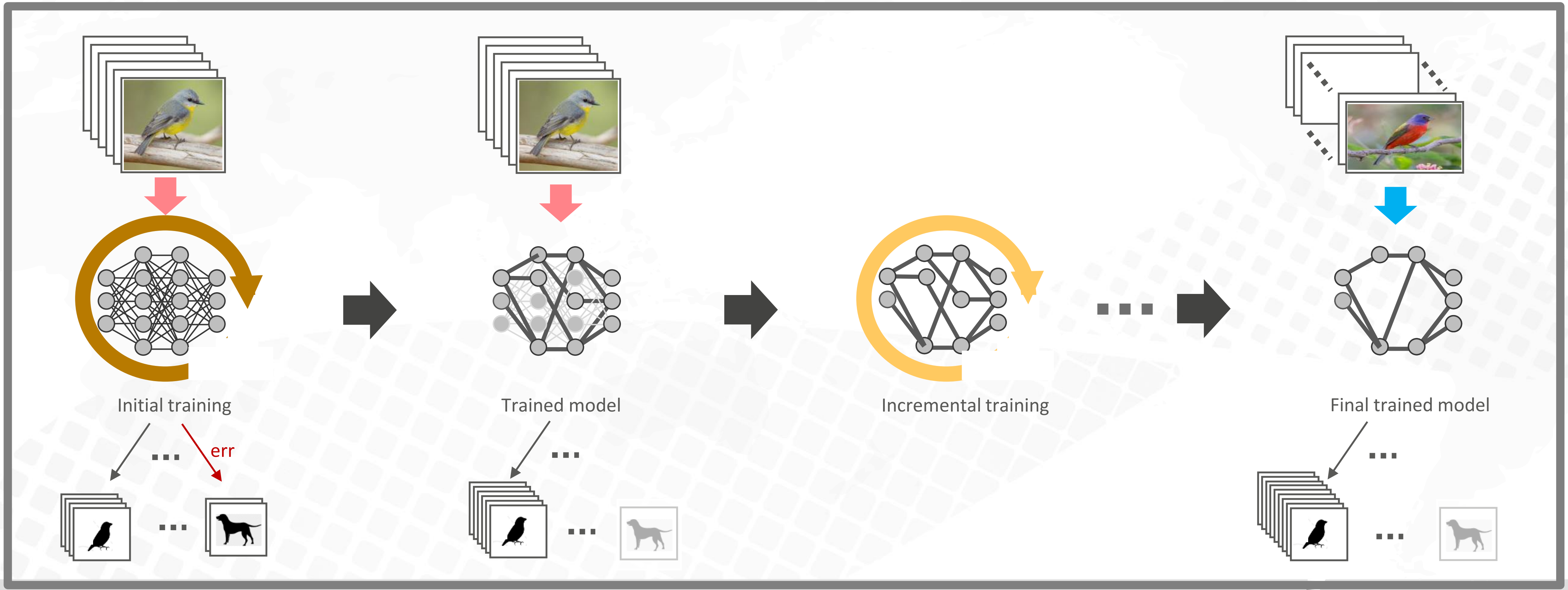
- Cloud Infrastructure Management Interface (CIMI) by Distributed Management Task Force (DMTF)
- Open Virtualization Format (OVF) specifications
- Organization for the Advancement of Structured Information Standards (OASIS)
- Cloud Application Management for Platforms (CAMP)
- OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA) interfaces
- Storage Networking Industry Association (SNIA) – Cloud Data Management Interface (CDMI)

Adaptive Model Training

- **Automated SSDC**
  - ✓ Adaptive configuration based on applications
  - ✓ SIMD (Single Instruction, Multiple Data) type of computations
  - ✓ Data reuse and stages pipelined computation
- **Domain specific AI machine learning**
  - ✓ Complicated network orchestration and heterogeneous environments
  - ✓ Optimized training with node pruning
    - Network device distribution
    - Adaptive server and CPU nodes
  - ✓ Automated application classification and network/DC configuration

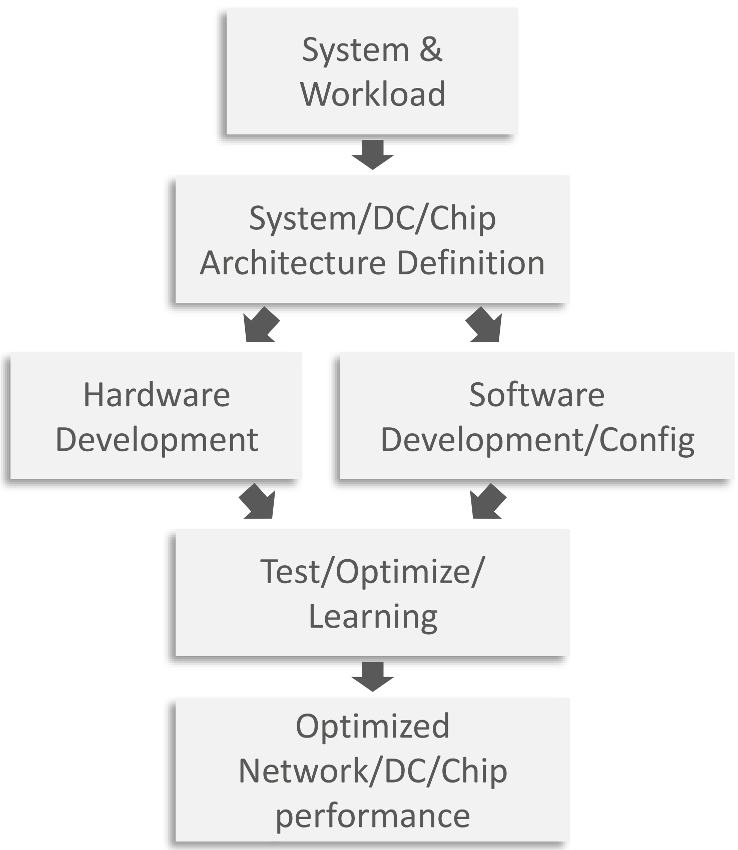
Adaptive Autonomous – Example

- Deep learning model training and optimization
  - ✓ Automated learning
  - ✓ Model pruning for optimization
  - ✓ Potentially reduced precision. Incremental training necessary



Hardware/Software Co-Design EDA

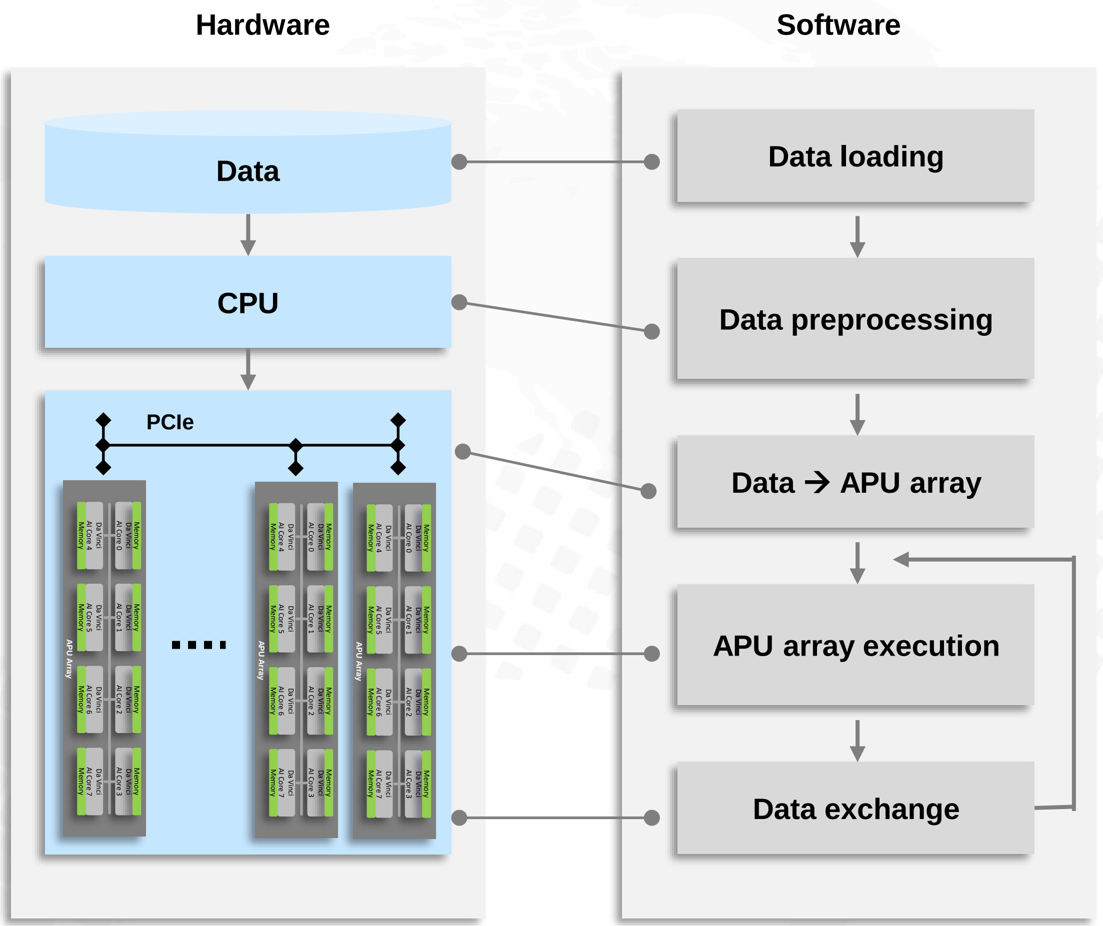
- **To improve the efficiency of software execution**
  - ✓ Hardware design should support the optimal software execution
    - Type of computation tasks – intent learning
    - Type of data and their size – structured data
    - Priority of latency, throughput, bandwidth – computation flow optimization
  - ✓ Server, CPU/APU, storage should be configured to best execute the tasks – intelligent configuration model
    - Twine models for simulation and learning
    - Dynamically adjust model based on continuous incremental learning
- **Hardware/software co-design EDA tasks**
  - ✓ Determine the hardware feature requirement and configuration
  - ✓ Build configuration model library and learning algorithm
  - ✓ Develop key KPIs for optimization measurement, such as various of workload scheduling measurement, throughput stages, etc.



Hardware/Software Co-Development Approach

Hardware/Software Co-Design EDA – Example

- **Co-design task and challenge**
  - ✓ Computation
    - Both maximal sharing and performance
    - Parallel, streamlined, Pipelined, asynchronous, etc.
  - ✓ Storage
    - Storage depth (size) and width (speed)
    - Pool storage vs synchronization/pipelining storage
  - ✓ Communication network
    - Between storage data
    - Between storage data and CPU
    - Between APU and storage data
    - Between APU and APU
- **Co-design library components**
  - ✓ Scalar for computation and storage
  - ✓ Vector macro for computation and memory
  - ✓ Repetition and branching timing and control
  - ✓ Etc.



FUTUREWEI Technologies

Hardware/Software Co-Design EDA – Analysis

- **Plan all types of tasks for deep learning models, such as**
  - ✓ Computation dominated
    - Top MLP
  - ✓ Communication dominated
    - Feature extraction and analysis
  - ✓ Memory bandwidth dominated
    - Bottom MLP
    - EMB lookup
  - ✓ Memory capacity dominated
    - Dense features
    - Sparse features
- **Benchmark model library**
  - ✓ Training and recommendation
  - ✓ Incremental collection and justification

Summary

- **Adaptive SDDC requires the supports from**
  - ✓ Hardware and software co-design
  - ✓ AI machine learning to automate the analysis of the intended tasks
  - ✓ Libraries for configuration models, machine learning models, execution framework and scalar, macro instruction set, etc.
  - ✓ Automated template recommendation including configuration and algorithm for service tasks
  - ✓ Twin model simulation and dynamical model adjustment
- **Community sharing**
  - ✓ Increase OCP availability
  - ✓ Advanced training model availability from various applications

Thank you

www.futurewei.com