Near Data Processing for AI & Big data
(Data Hierarchy)

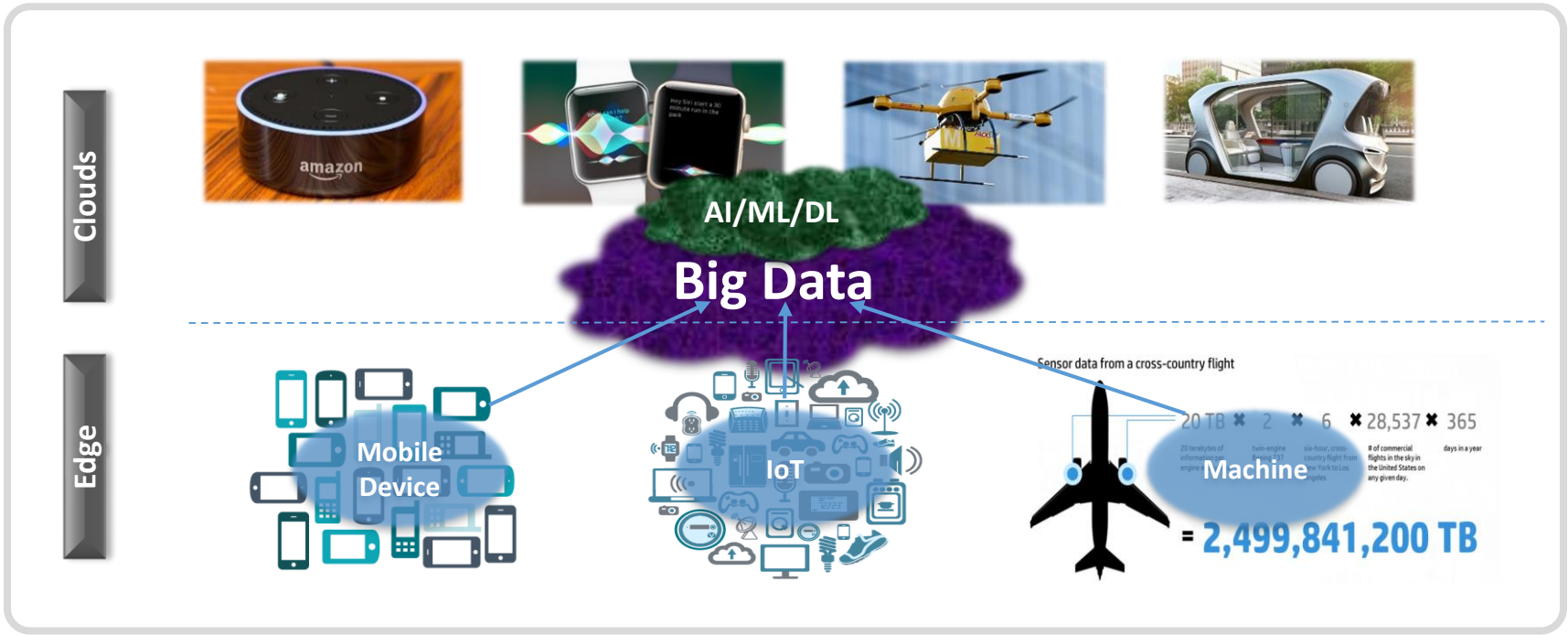
SK hynix

MEMORY SYSTEMS R&D Eui-Cheol Lim

Contents

AI and Big Data

- AI with big data processing is one of the key technologies in modern industry
- From the edge to the clouds, AI is proven to be advantageous in various application fields



Computer vs. Human Brain

- AI has already proven that it is more capable than human in certain applications
- However in energy perspective, human brain is much more efficient than current computer system

Go match : AlphaGo vs. Lee Sedol

AlphaGo
1202 CPUs, 176 GPUs, 100+ Scientists
170kWh

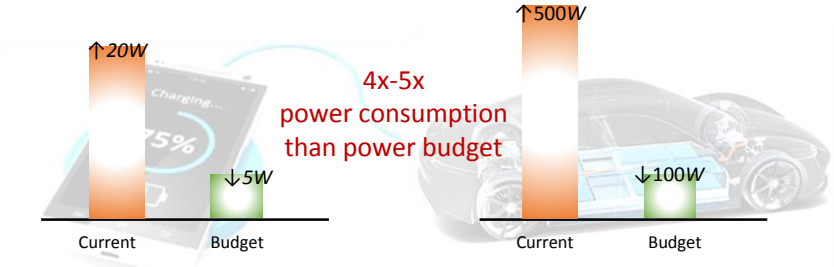
Lee Sedol
1 Human Brain, 1 Coffee
20Wh

Power Efficiency – Challenges of AI

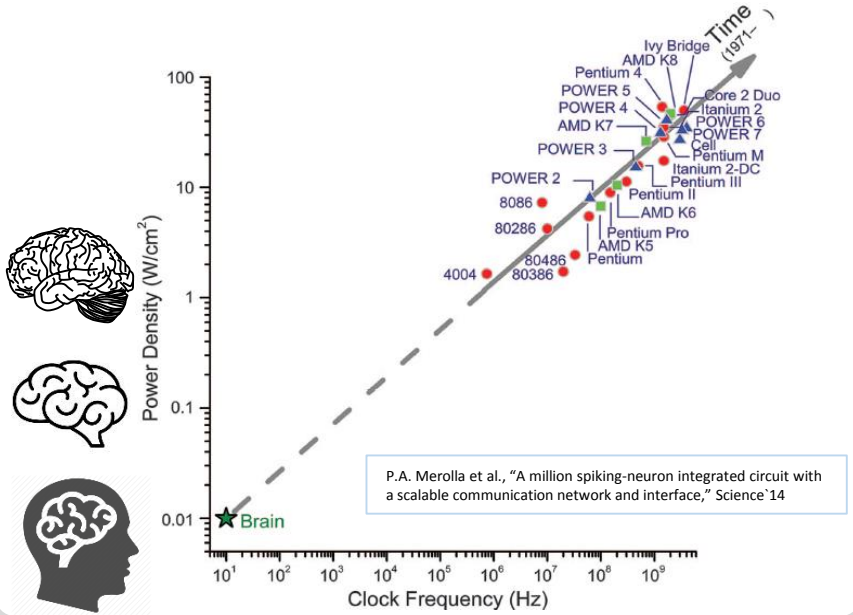
- **AI performance is limited by power budget**
- **Power density and clock frequency of traditional computer architecture is constantly increasing while brain operates at the lowest point**

Power Limitation in AI application

- Current AI applications are power hungry
  - Most of AI apps already exceed its power budget
  - AI still needs more data to enhance accuracy, and current architecture cannot satisfy that requirement



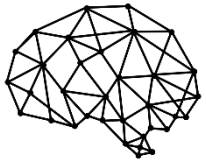
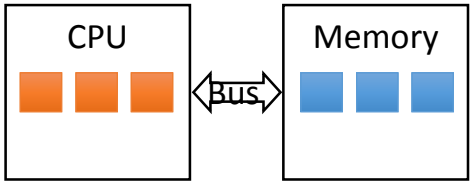
Power Efficiency: Human Brain vs. Computer



Contents

Human Brain vs. Neuromorphic vs. von Neumann

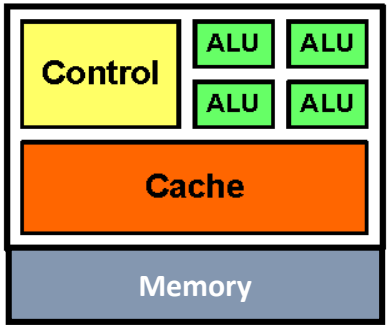
- **Neuromorphic computing looks far better than von Neumann architecture, but it is in premature stage yet**

	Human Brain	Neuromorphic	von Neumann
Architecture			
Device	Neuron / Synapse	Artificial Neuron / Synapse	CPUs / Memories / Bus
Features	Event-driven	Event-driven / Asynchronous	Computing-centric / Compute, memory separated
Encoding Scheme	Spiking Signals	Analog/Digital/Mixed Spiking Signals	Binary Signals
Accelerator	-	Neuromorphic Processor	GPU / FPGA / ASIC
Power / Energy Consumption	Ultra Low	Low	Relatively High

Architecture Revisited: von Neumann

- **Data movement is a common issue in von Neumann architecture and causes power efficiency issue as AI requires big data calculation**

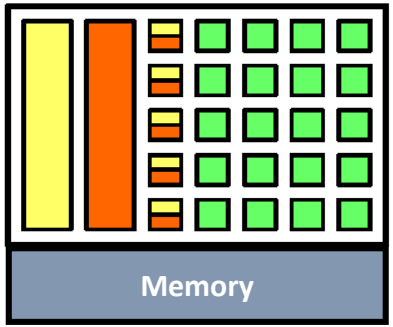
CPU based



- Integrated multi-core

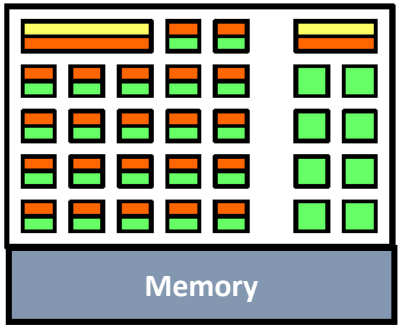
→ Fit for complex problem

GPU



- Thousands of parallel computing unit
- Simple & Massive computing

NPU (ASIC/FPGA)



- $< 10^4$ parallel processing element
- PE array for matrix multiplication

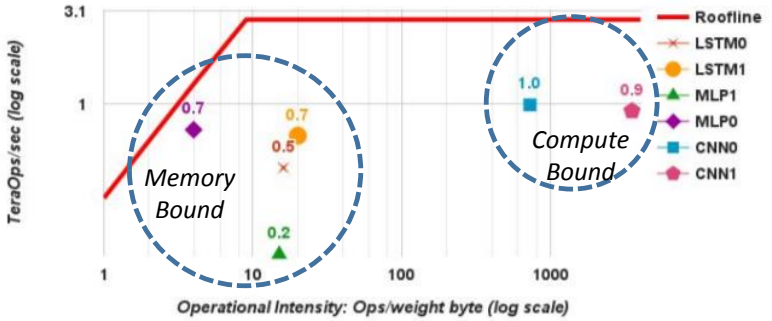
→ Optimized for NN

Energy Cost of Data Movement

- **Most AI algorithm is memory bandwidth bounded**
- **Data movement consumes more than 60% of power in modern processors**

ROOFLINE Graph According to Neural Network

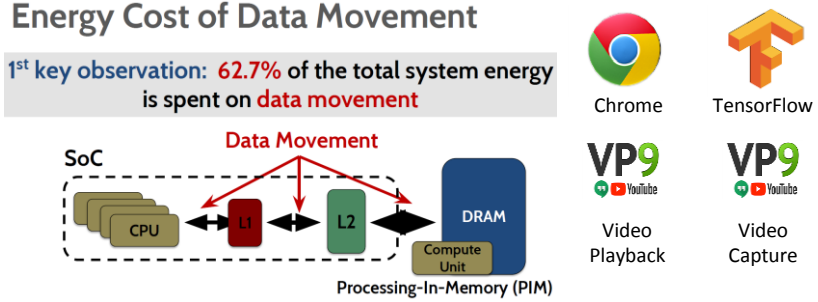
- In most cases, NN performances are bounded by B/W
  - Recent NN Accelerators use internal memory to reduce bottleneck, but it is not sufficient.



Norman P. Jouppi, *"In-Datacenter Performance Analysis of a Tensor Processing Unit,"* ISCA'2017

Power Portion by Data Movement

- More than 60% of power is consumed by data movement
  - Analyze power consumption in popular google apps

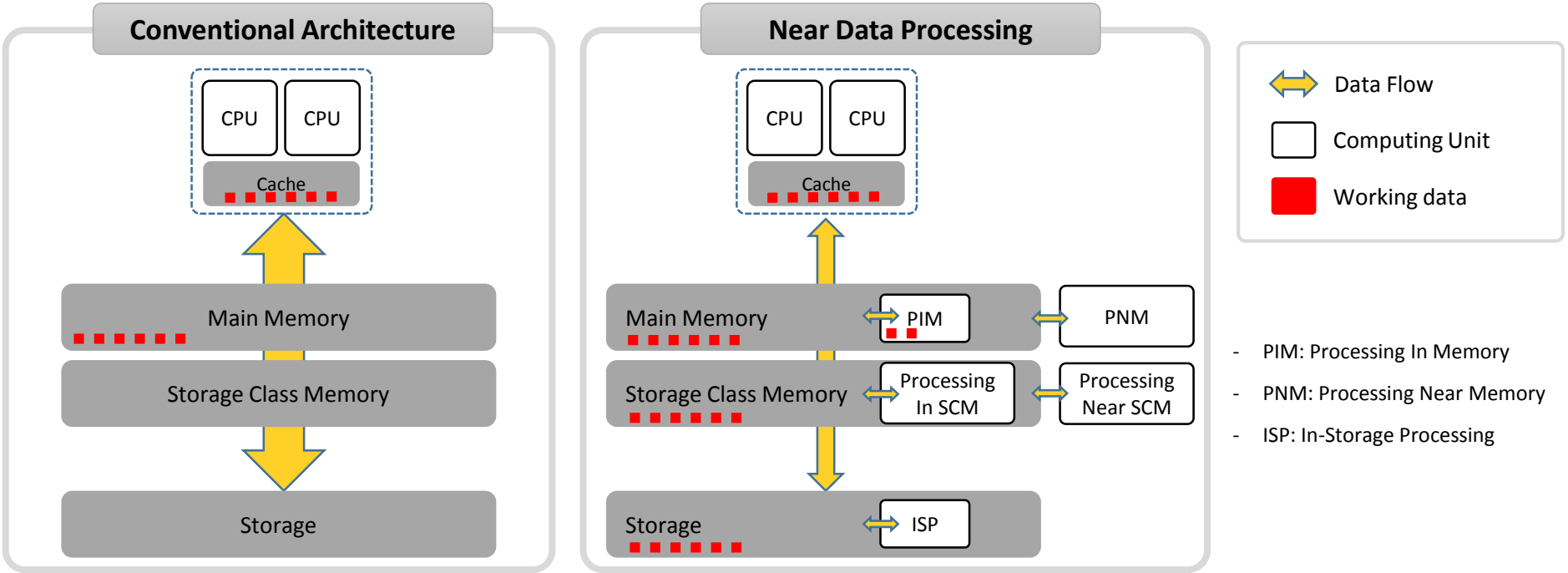


Amirali Boroumand, *"GoogleWorkloads for Consumer Devices: Mitigating Data Movement Bottlenecks,"* ASPLOS`18

Contents

Moving Computation To Data

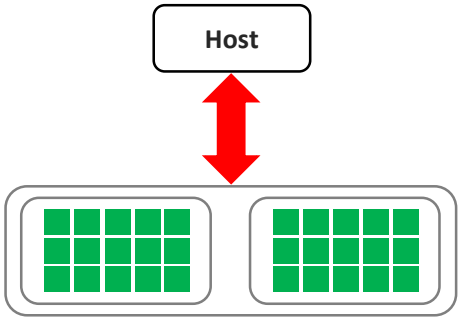
- **Near Data Processing can reduce energy for data transfer by locating compute node where data lives**



Near Data Processing Concept at Memory Level – PIM, PNM

- **PIM (Processing In Memory), PNM (Processing Near memory)**
  - **Pros: Overcome bandwidth limitation between logic and memory devices with lower power consumption**
  - **Cons: Not backward compatible with legacy software stack, requiring some changes in system software**

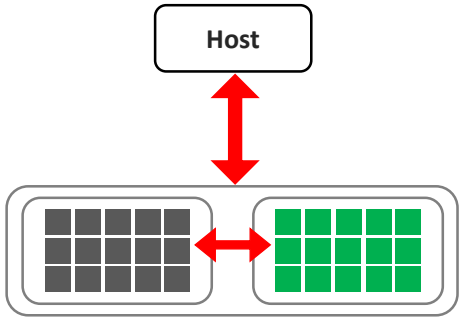
Conventional System



✓ SW changes: none

✓ Power consumption: relatively high

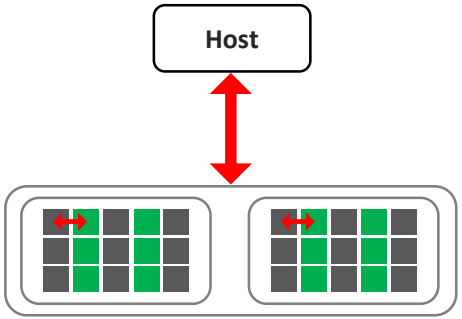
Processing Near Memory



✓ SW changes: **small** or **moderate**

✓ Power consumption: relatively low

Processing In Memory



✓ SW changes: huge

✓ Power consumption: **very low**

: Computing

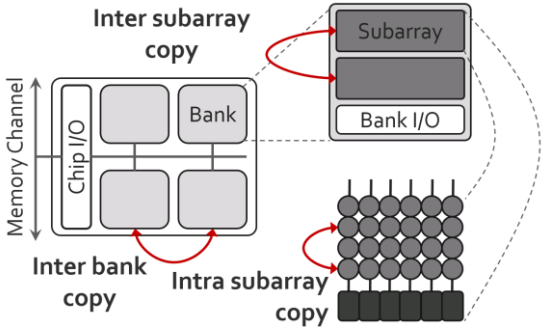
: Memory

PIM, PNM Researches

PIM

- **Row data copy**

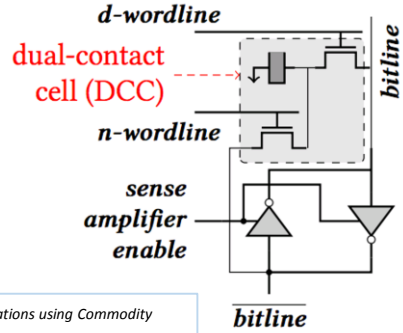
e.g.) Bulk copy



Seshadri et al., "*RowClone: Fast and Energy-Efficient In-DRAM Bulk Data Copy and Initialization*," MICRO`13

- **Bulk bitwise operations**

e.g.) In-DRAM bit-wise NOT

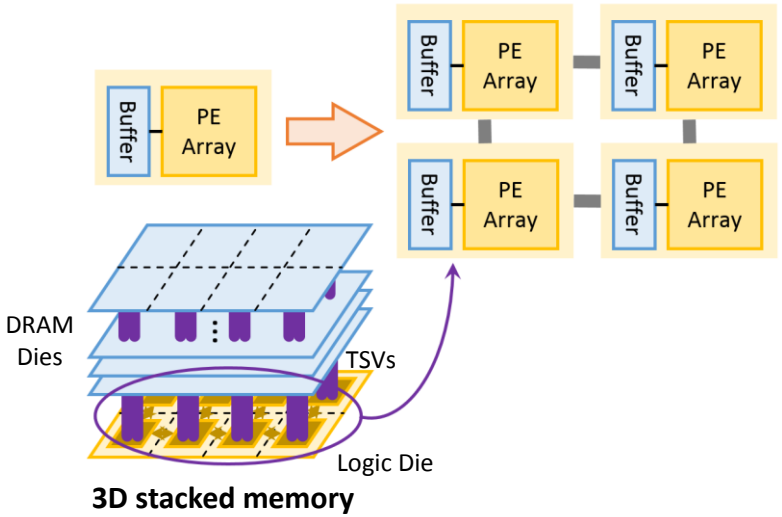


Seshadri et al., "*Ambit: In-Memory Accelerator for Bulk Bitwise Operations using Commodity DRAM Technology*," MICRO`17

PNM

- **3D stacked Memory**

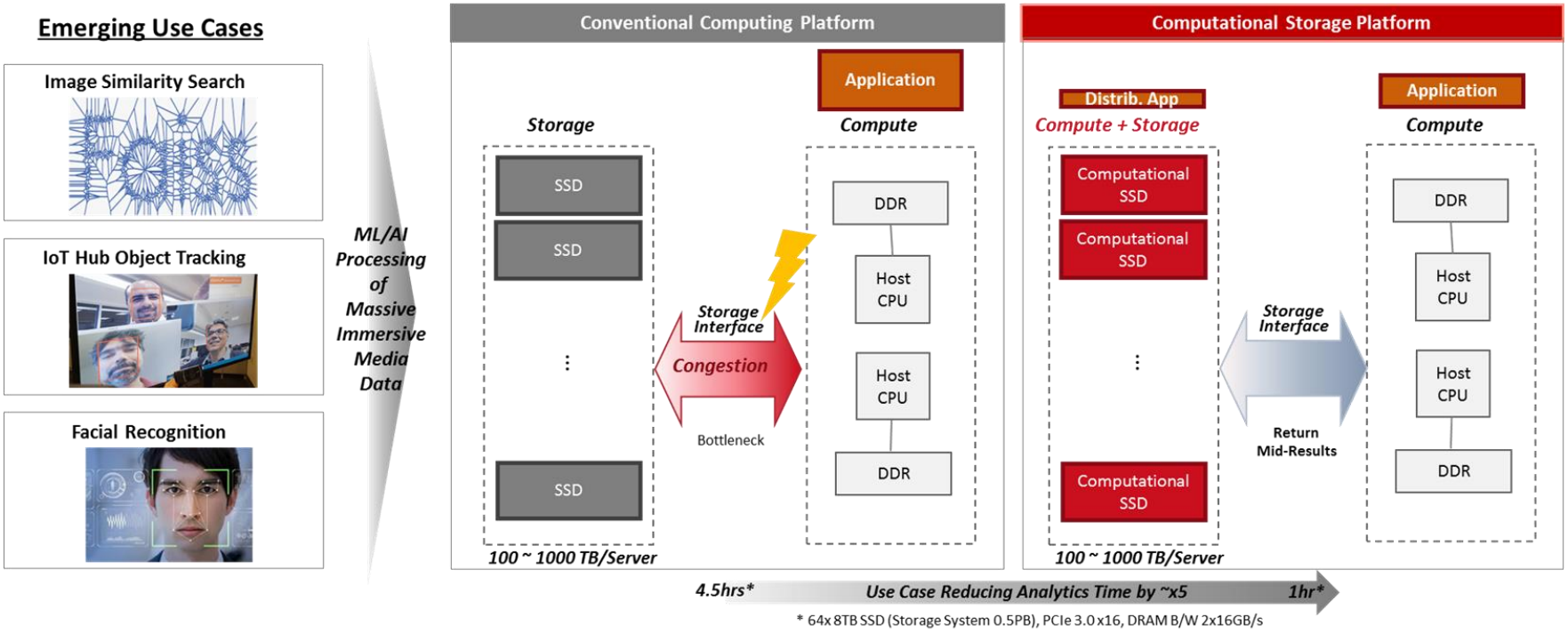
e.g.) NN accelerator



M. Gao et al., "*TETRIS: Scalable and Efficient Neural Network Acceleration with 3D Memory*," ASPLOS`17

Near Data Processing at Storage Level – In Storage Processing

- **As ML/AI based Big Data analytics workload increases I/O congestion between storage and compute node which could cause a performance bottleneck**



Researches on In Storage Processing

- **Big-data analysis, distributed machine learning are key target workloads for computational SSD researches**

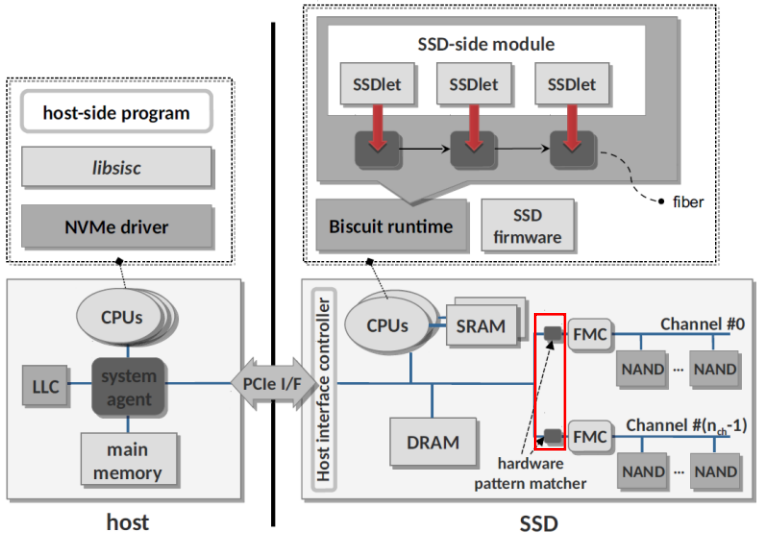


Figure 2. Overall architecture of a Biscuit system.

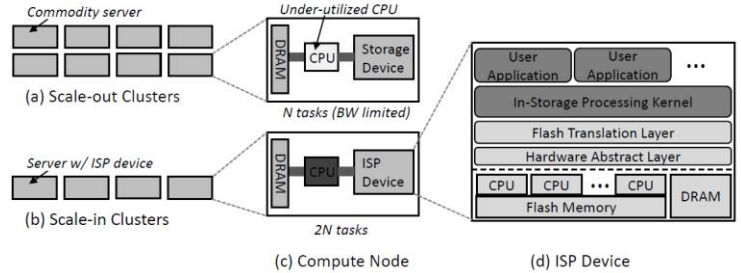


Figure 5: Scale-in clusters with ISP. ISP devices can be implemented utilizing modern SSD architectures.

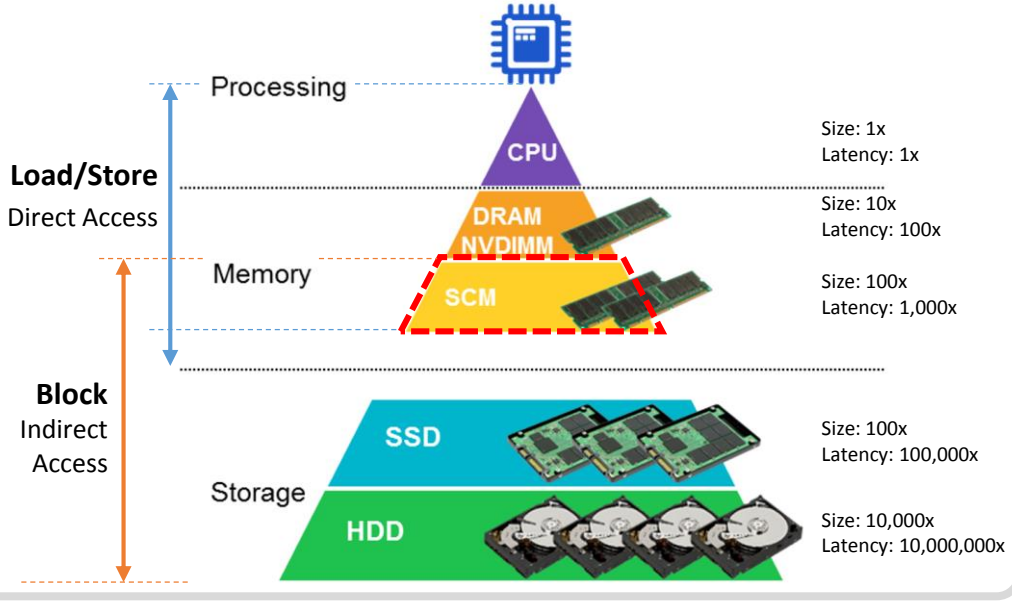
I. Choi et al.,"*Energy efficient scale-in clusters with in-storage processing for big-data analytics*," MEMSYS`15

B. Gu, et al., "*Biscuit: A Framework for Near-Data Processing of Big Data Workloads*," ISCA'16

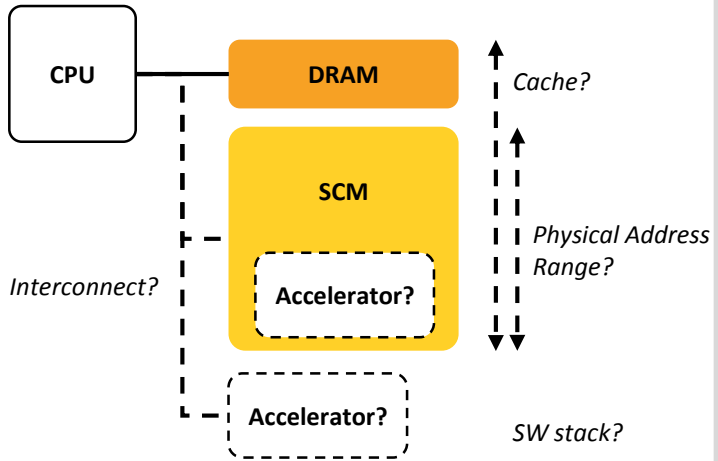
Near Data Processing for SCM/NVM Layer

- **SCM (Storage Class Memory) could be useful in increasing the memory capacity: slower but larger, cost efficient compared to DRAM, and persistent**
- **SCM utilization models are actively being researched including the application in the NDP structure**

Memory Hierarchy



SCM Utilization Model



Source: Rambus

Why Near Data Processing Again?

- **Emerging applications' strong demands are driving Near Data Processing Architecture**
- **Lower hanging fruit – 3D, 2.5D stacking technology improves feasibility of Near Data Processing**

Strong Demands

General application

- Computing centric workloads
- High locality, data reuse

Emerging big data, AI applications

- Memory-intensive workload
- Low locality, abundant parallelism
- Bandwidth & Energy Constraints
- Irregular access patterns

Feasibility

High cost of integrating compute units within DRAM

3D, 2.5D die stacking

- Low cost
- High feasibility

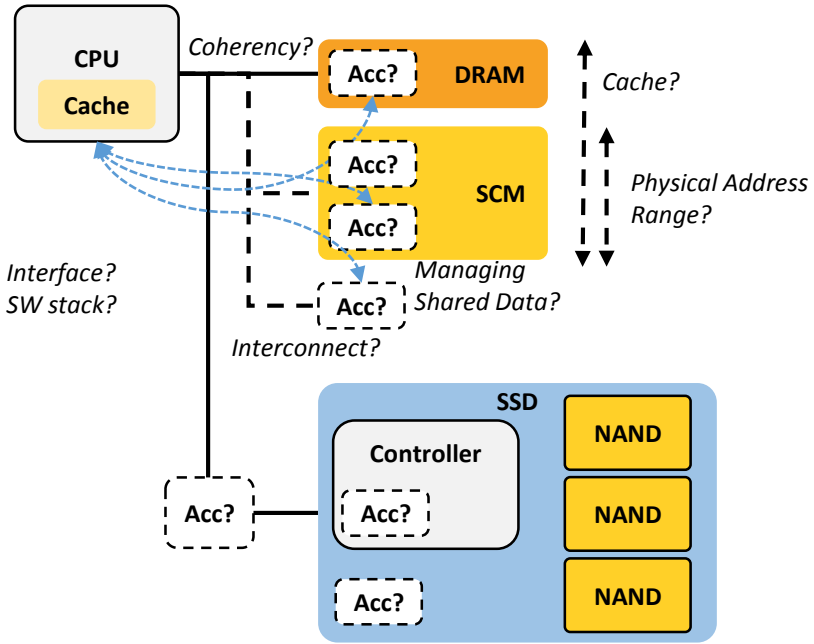
than computing in DRAM

Key challenges in NDP - System

- **Many challenges exists in enabling near data processing technology to be widespread**

System Issue

- How to design **novel accelerator architecture?**
- How to support **virtual memory space**?
- How to maintain **cache coherence**?
- **Parallel computing** scheme?
  - e.g.) How to manage shared data
- How to design **data structure** for NDP?
- What kind of **interconnect** should be used?

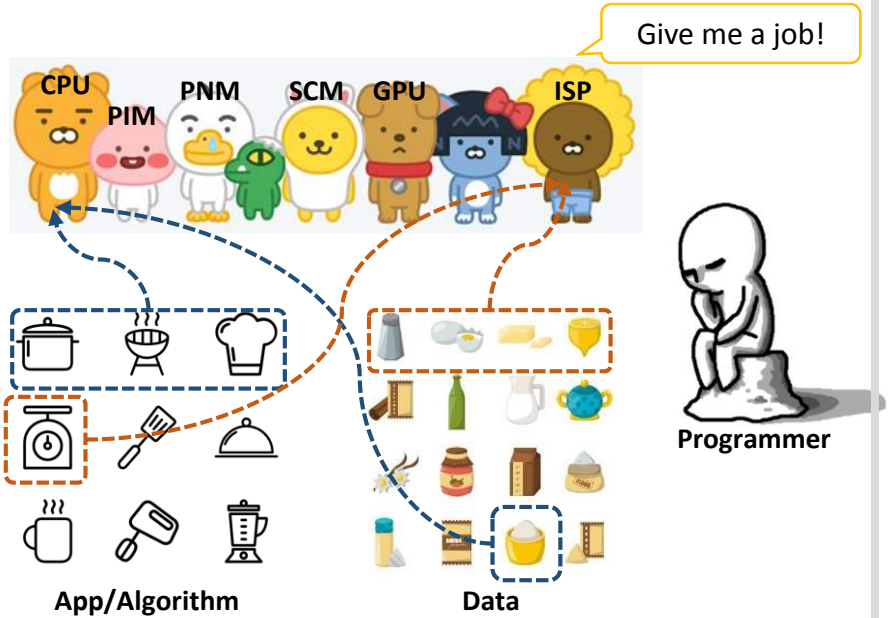


Key Challenges in NDP – Programming Model

- **How to establish a programming model is also an important issue**

Programming Model

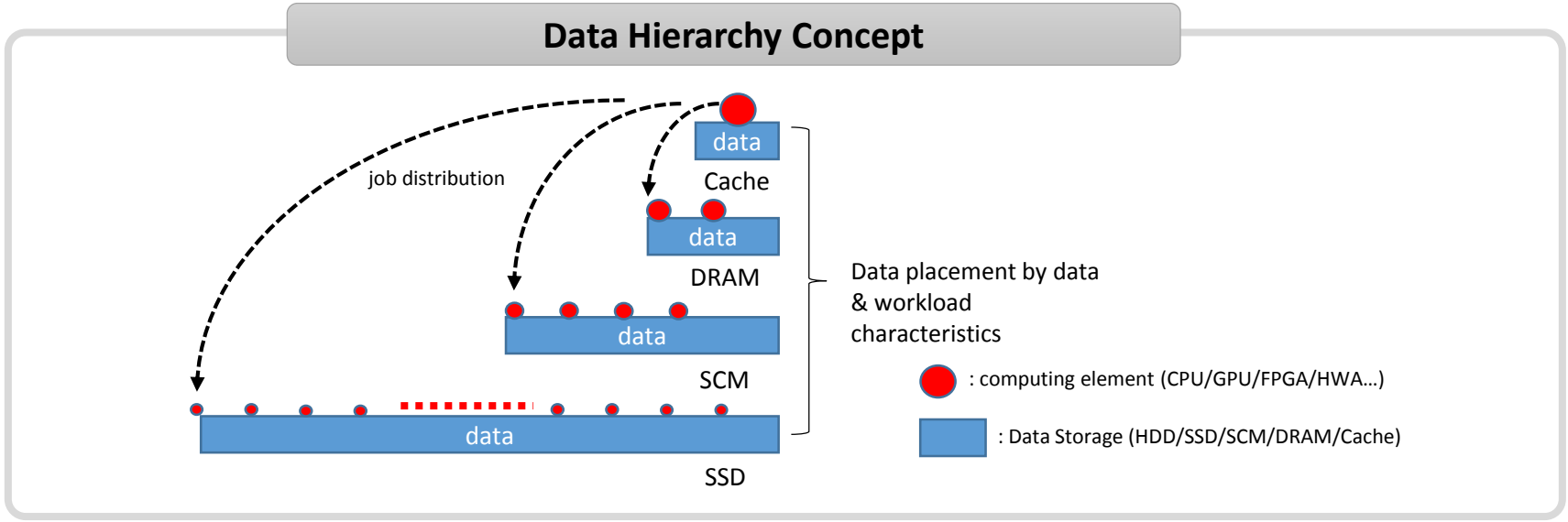
- **Which part** of an application should run in the NDP units?
- **How** to schedule code? Static? Dynamic?
- **Who** schedule? Programmer? Compiler? OS?
- Need to **integrate** system software with programming frameworks



Contents

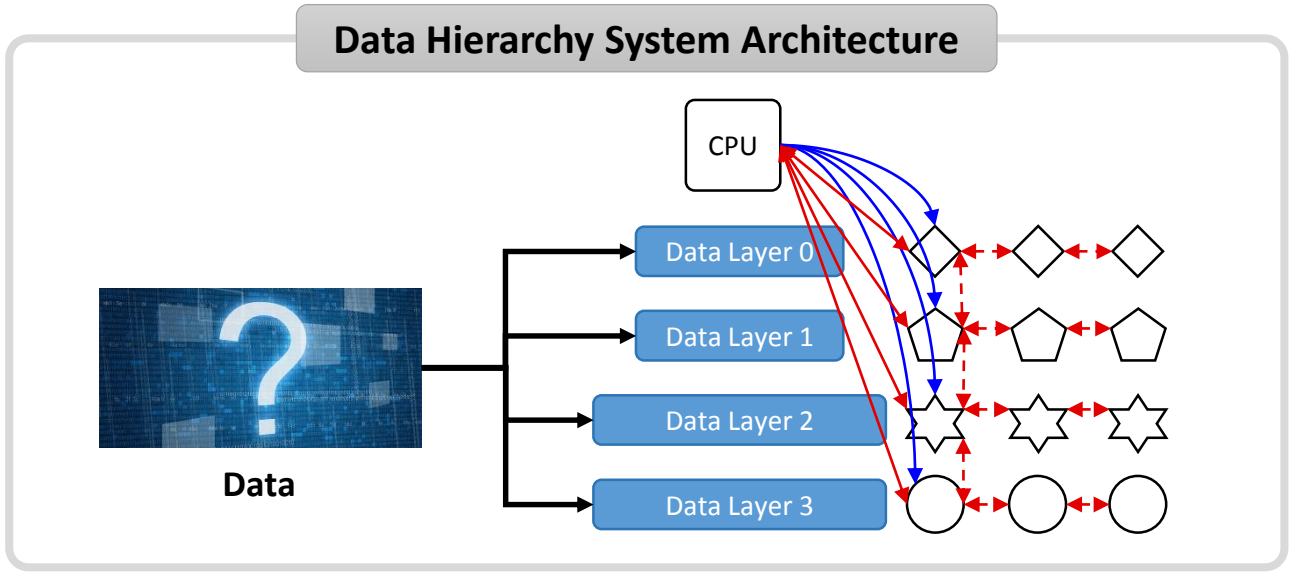
Data Hierarchy – NDP For All Memory Layers

- **Basic Concept – Every Memory layers have own processing element**
  - Minimizing data movement for the **entire memory hierarchy**
  - **Data placed** based on the data & workload's characteristics
  - **Compute data where data reside**
- We name this system architecture as "**Data Hierarchy**"



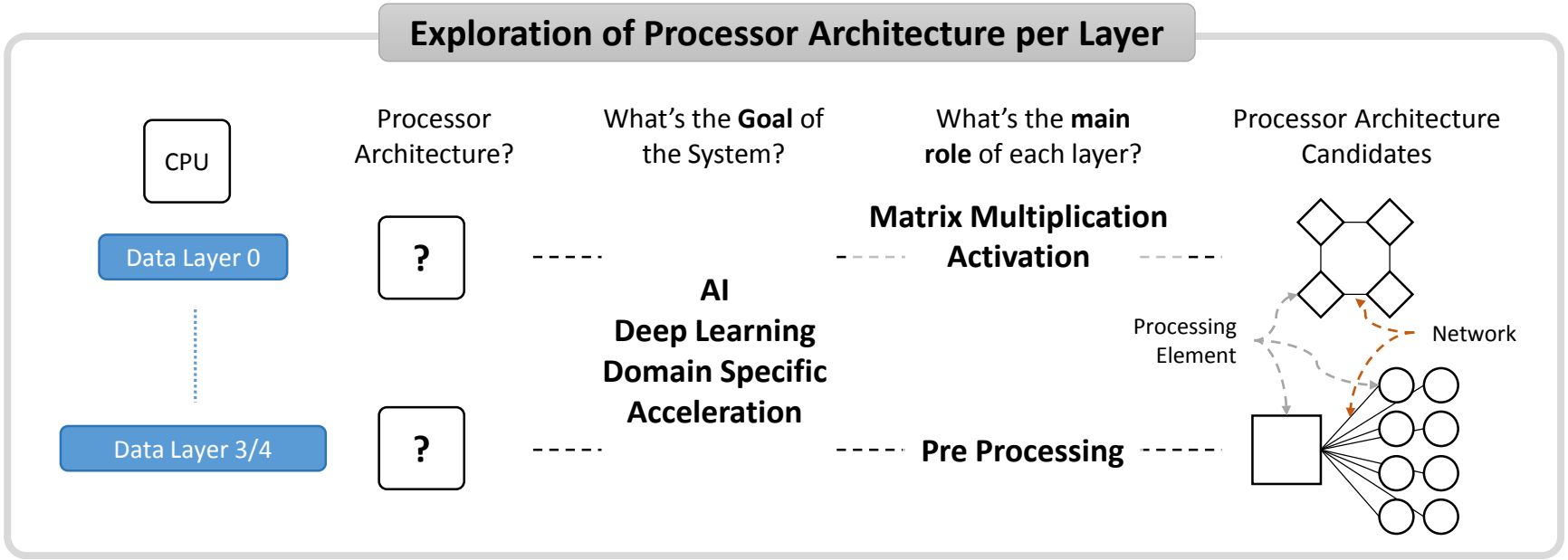
Research Topics in Data Hierarchy – Data placement

- **Data placement**
  - How to **classify** the characteristics of data?
  - How to **place** the data at the appropriate layer
  - Whether the characteristics of data will be **determined only by the data itself**, or **by the algorithms** that utilize the data



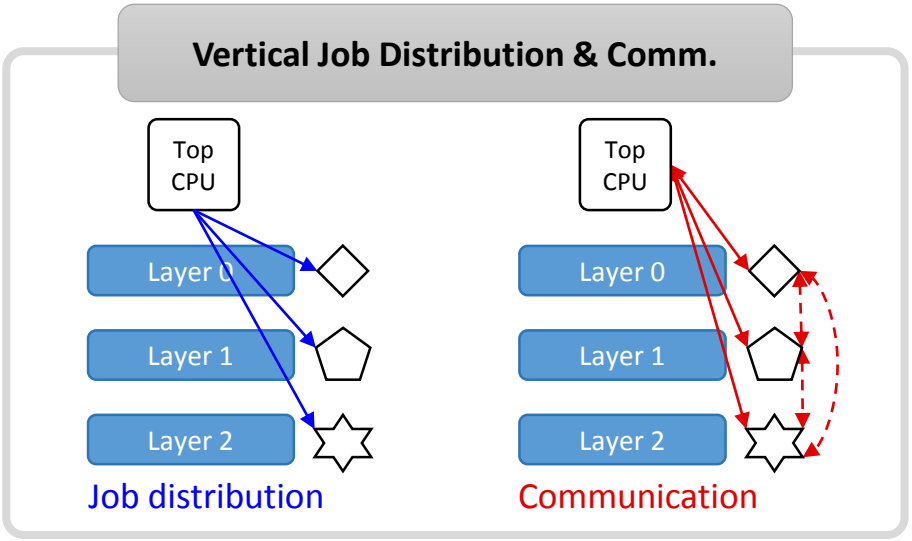
Research Topics in Data Hierarchy – Processor Architecture

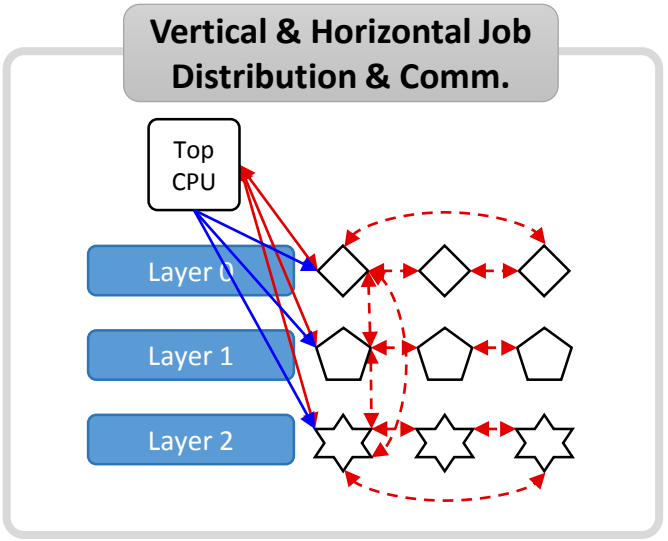
- **Processor architecture per layer**
  - What is the feasible structure of processing elements for each layer?
  - **Domain specific architecture** is the key



Research Topics in Data Hierarchy – Job Distribution & Comm.

- **Job distribution & Communication**
  - In terms of job distribution, Data Hierarchy is similar to heterogeneous computing
  - Framework is required to assign a job to each layer and aggregate the results





Contents

Conclusion

- As AI and Big-data became widely spread, system architecture has to be improved to solve energy efficiency and performance scaling issues
- One of the ways to solve the issues is to place the data in different layers based on characteristics and processing the data independently within that layer → Data Hierarchy Concept
- Yes, there are lots of research topics for Data Hierarchy
  - Data placement, data mapping
  - Processor architecture for each layer
  - Job distribution

"Start from simple and domain specific applications, but have to target as general as possible"

THANK YOU

E-mail: euicheol.lim@sk.com

SK hynix MEMORY SYSTEMS R&D