April 26-28, 2022
DoubleTree by Hilton San Jose
SmartNICsSummit.com

Fourth Generation Architecture for SmartNICs

Jim Finnegan & Niel Viljoen

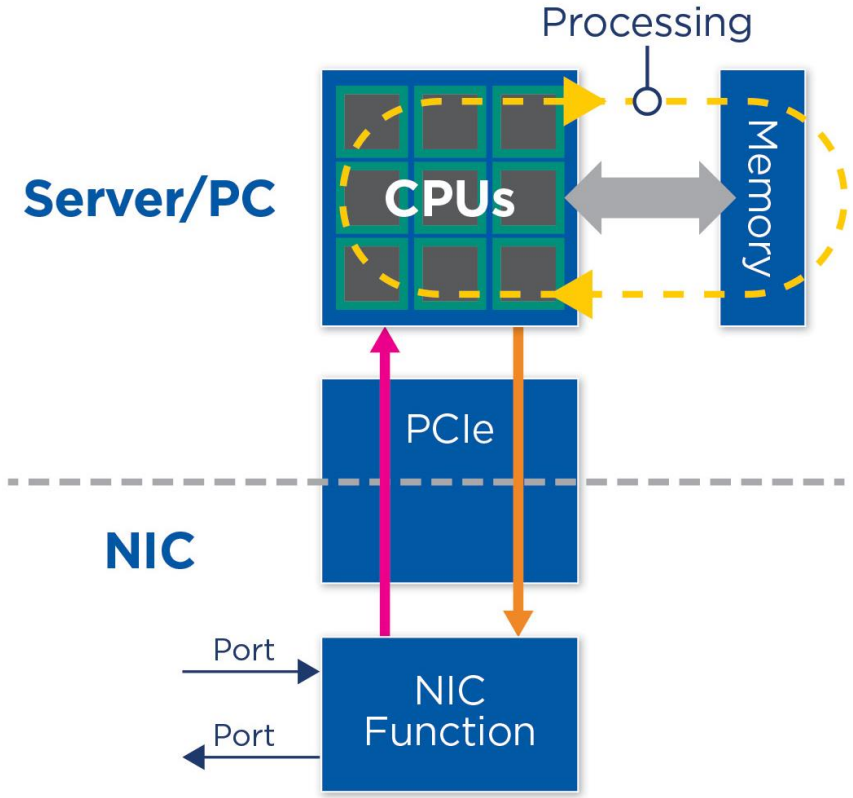
"Conflicting Requirements" conundrum

Generalized intent of SmartNICs is to "*offload certain functions from host*"

- ❑ how can this be accomplished cost-effectively? (review the evolution of implementation options)
- ❑ how can the offloaded functions be flexibly adjusted? (implies software programmability)
- ❑ how can the solution address exponentially increasing bandwidth link speeds? (implies the need for scalability)
- ❑ how can the implementation leverage and incorporate domain specific elements? (implies adoption of open standards and 3rd party integration)
- ❑ how can the solution meet the power-performance-area dichotomy? (implies cognizance of Dennard Scaling and Moore's Law through use of chiplets)

First Generation ~~Smart~~NICs

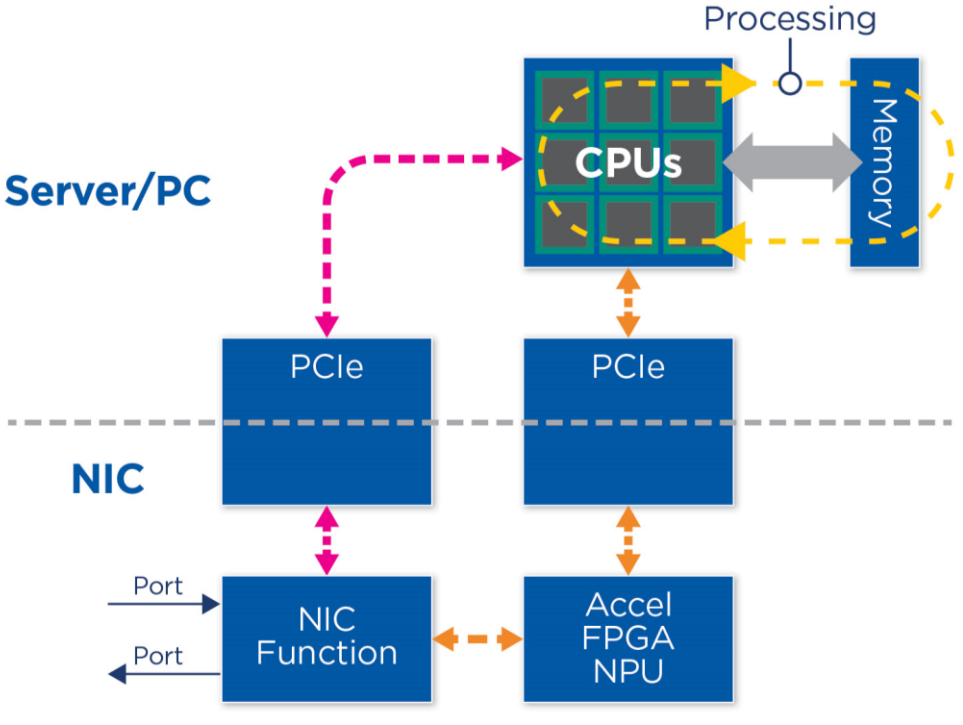
- First generation SmartNICs (sic) offloaded rudimentary stateless functions such as CRC, RSS, etc.
- TOE was first example of stateful offload
- Most of the network processing was still done in the host



- **1st generation referred to as "BasicNIC" or "CoreNIC"**
- **Rudimentary offloads did NOT comprehend *state***
- **These NICs did not have intelligence to send pkts to egress port without traversing PCIe (perf. impact)**
- **Historical footnote – TCP/IP temporarily overshadowed in 1990s by jitter tolerant ATM (Asynchronous Transfer Mode)**

Second Generation SmartNICs

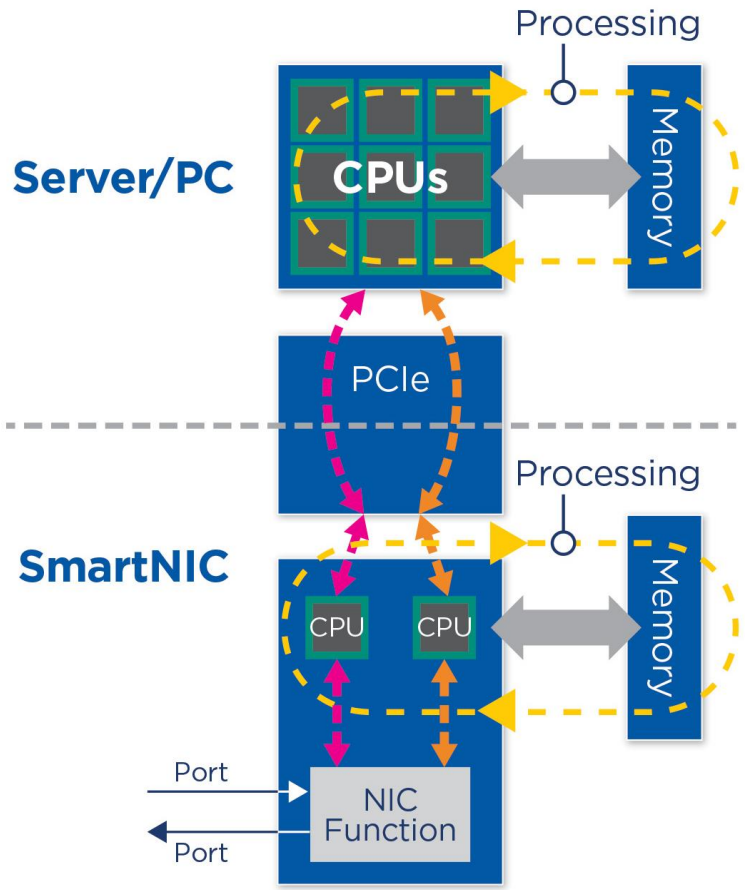
- Aka “Offload-” / “Feature-” NIC
- Acceleration can be in-line or side-looking
- Accelerator can add Meta-data to packets if in-line
- NIC function needs multiple interfaces to classify and steer traffic
- FPGA or NPU may additionally offload selected NIC functions



- **2nd Generation generally referred to as “Offload NIC”**
- ***State* processing added via NPUs or FPGAs**
- **Classic example is Microsoft’s GFT (Generic Flow Table) offload (30% of host processing)**
- **Challenge is the X18 factor -Profs Kuon & Rose seminal book “Quantifying and exploring the Gap between FPGAs and ASICs**

Third Generation SmartNICs

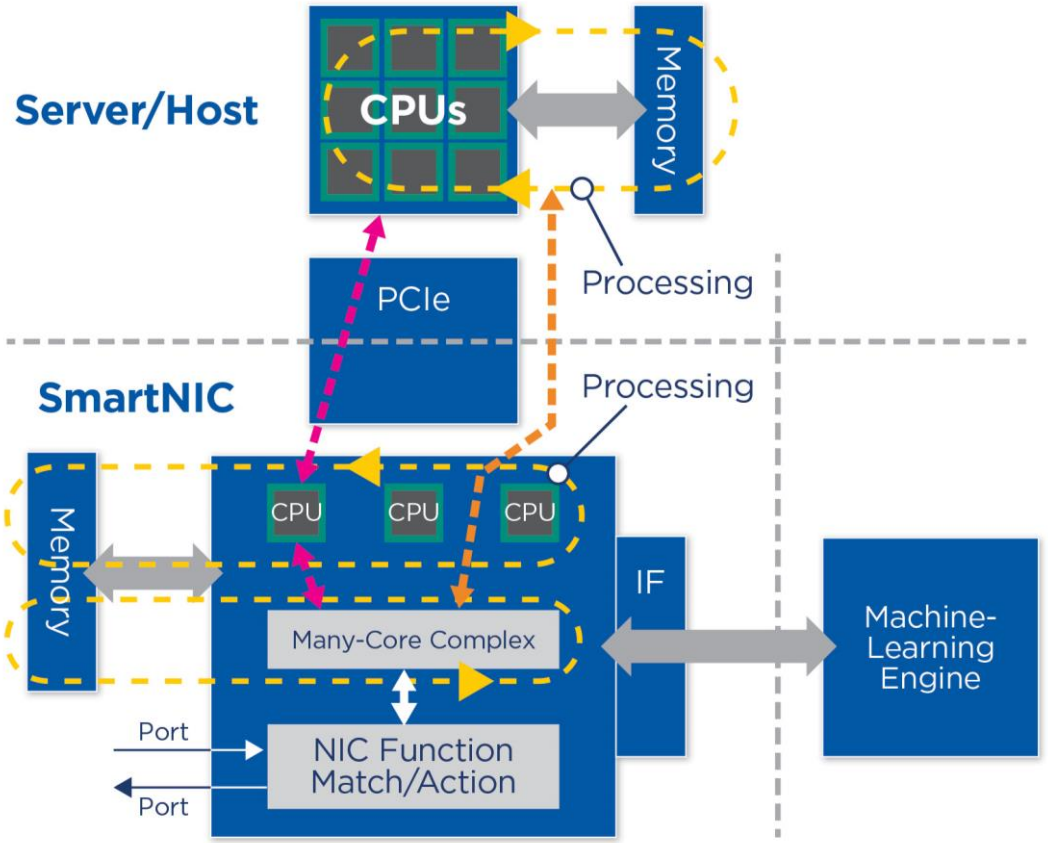
- General-purpose cores allowed migration of server/host SW onto similar SMP (Linux) cores
- Dedicated Memory on NIC for cores
- Offload was typically OVS and other Virtualized Network Functions (VNFs)
- Problem remains the amount of packet processing SMP cores can do
- Standard NIC function with weak classification ability and little/no programmability was used. (However, some devices with P4-like Match / Action were also deployed)



- **The 3rd generation flexibility often used general purpose CPU complex**
- **Hence flexible datapath / some persisted with FPGAs….**
  - RTL versus C factoid; 4.5 million software engineers in US versus 80K computer hardware engineers
- **Recall Amdahl "Rule of thumb"**
  - 1Gbps I/O requires 1Ghz processing
- **Consider Ethernet Forum Roadmap**

Fourth Generation SmartNICs

- Link speeds increased to 200-400 Gbps
- Allow more stateful handling at lower levels
- Support external ML/storage acceleration without server involvement
- Customers need (proven, open) SW programmability and hardware flexibility ➔IP/chiplets



- **Bandwidth dictates datapath imperative using MIMD (Multiple Instruction Multiple Data) design with large number of small cores in "run-to-completion" model**
- **Three-tier processing model**
- **Cost (Power-Performance-Area) imperative leads to chiplet implementation**

Summary and Conclusions

Key attributes of fourth generation smartNIC include:-

- **Programmability** – many open cores organized as MIMD complex
- **Flexibility** – disparate chiplets from domain experts (I/O chiplet, processor chiplet, ML etc.)
- **Composability** – heterogeneous 3rd party elements, different foundries, re-use
- **Scalability** – consistent scalable architecture from 25G to nX100G, without blocking or packet loss (including very large packets)
- **Power-Performance-Area-Yield** – Dennard Scaling and Moore's Law limitations dictate that a chiplet implementation is superior to a monolithic implementation

Axioms/Truisms:

- **Amdahl's Rule of Thumb *("1Gbps I/O requires 1Ghz Processing")***
- **AMD EPYC™ analysis ("the cost of a quad chiplet design is only 0.59 times the cost of a monolithic approach despite consuming 10% more silicon")**
- **Professor Rose's seminal book – *Quantifying and Exploring the gap between FPGAs and ASICs* ➔ X18**

Thank You

- Questions