Exploiting Network-Near Processing: From Smart NICs to Information Appliances

Ada Gavrilovska
CERCS
Georgia Institute of Technology

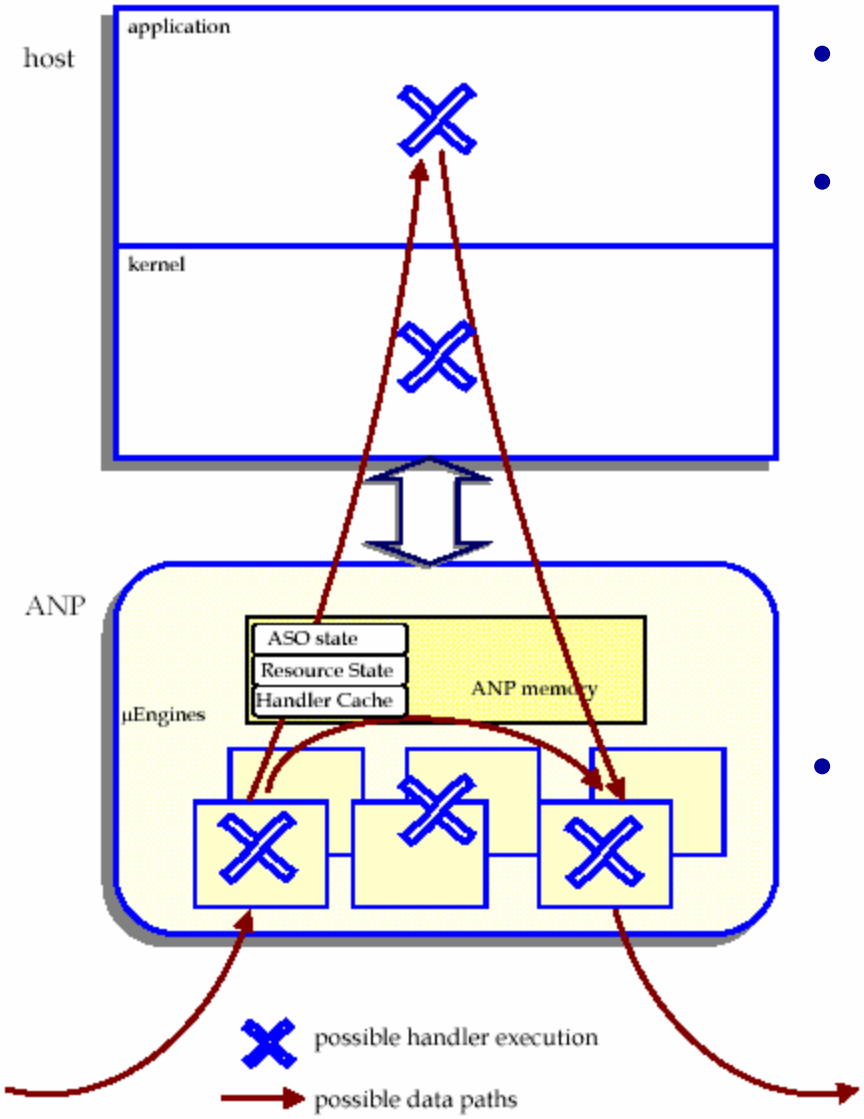
‘Network-near’ Processing Cores

- Existence and openness of processing cores dedicated to and/or specialized for communication tasks
  - Smart NICs
  - Network processors
  - Cores (or configurable accelerators) in multi- and many- core systems
- Benefits include
  - greater concurrency and overlap of computation and communication
  - reduced perturbations and memory and IO pressure on `main’ application components
  - specialized cores have reduced power requirements, optimized ISAs for network operations, `cheaper’ access to network packets…
  - lightweight or lack of OS -> lack of certain overheads

Multiple efforts

- Primarily based on Intel IXP network processors as development platform
  - as standalone systems, future NICs attached to general purpose hosts, or considered jointly with host CPUs as heterogeneous multicore platforms
- Addressing *higher-level services*, instead of just traditional network-level header and payload manipulations
- Three separate efforts
  - Dedicated cores
    - focus on systems services, e.g., IO acceleration and virtualization (Himanshu Raj and Sanjay Kumar presentations)
    - Intel funding
  - Specialized communication subsystem, i.e. NIC
    - ongoing work on application/middleware stack splitting on hosts with `attached NPs' acting as NICs
    - High-end SmartNIC targeting HPC systems
    - joint with RNet, Dept. of Energy SBIR II funding
  - Specialized platforms for customized communications-services
    - Application-level information appliances
    - Intel funding

‘In-transit’ Data Transformations

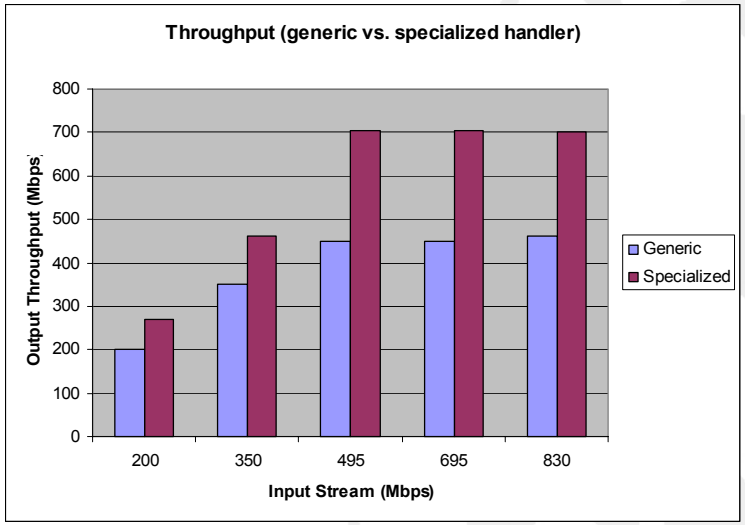


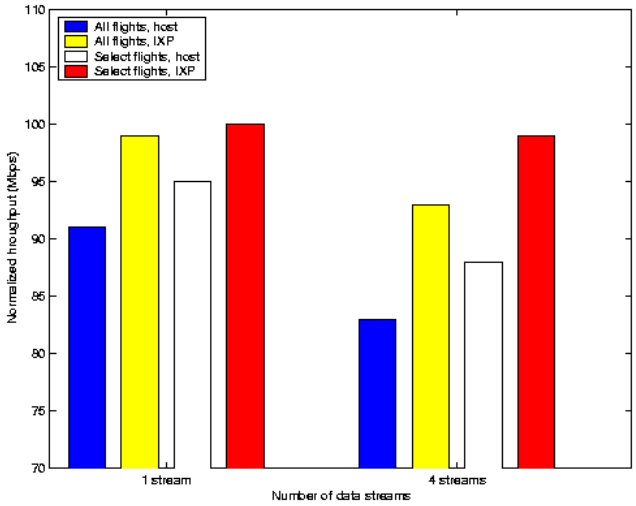
- vertical movement of service components along application stack
- enable execution of middleware-/application-level processing actions jointly with communications
  - use metadata to describe application data, processing actions & requirements, platform state…
  - offload computational CPU, enable direct data placement of *needed* data
  - deploy computation (handlers) onto contexts best suited for its execution
- configure paths dynamically based on application needs, context capabilities…
  - flexible classification
  - runtime monitoring
  - dynamic code deployment/selection
  - `cache’ current state

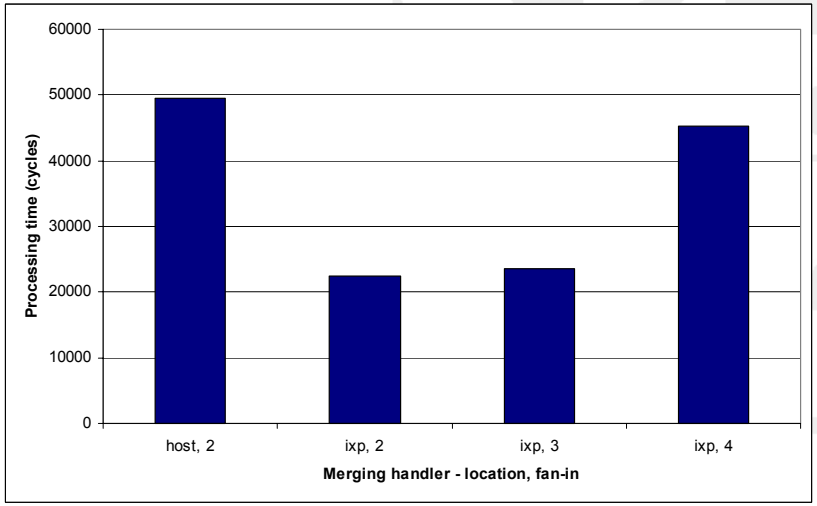
Application-specific Appliances

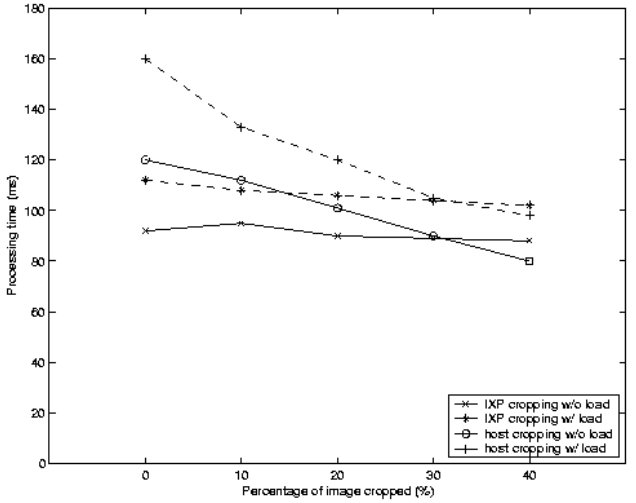
- Open platform to consolidate rich set of services
  - content distribution and filtering, XML translation and data translation, data integration…
- Flexible scheduling and resource allocation subsystem to meet dynamic range of QoS levels
  - dynamic allocation of processing context to flows
- Analyze impact of state and data placement along memory hierarchy
  - code specialization as topmost level
- Karsten Schwan, Sanjay Kumar, Radhika Niranjan, …

Select experimental results





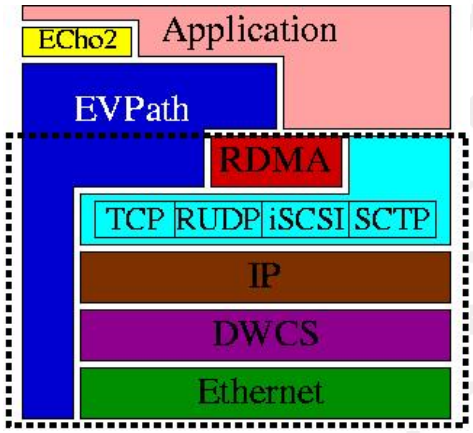


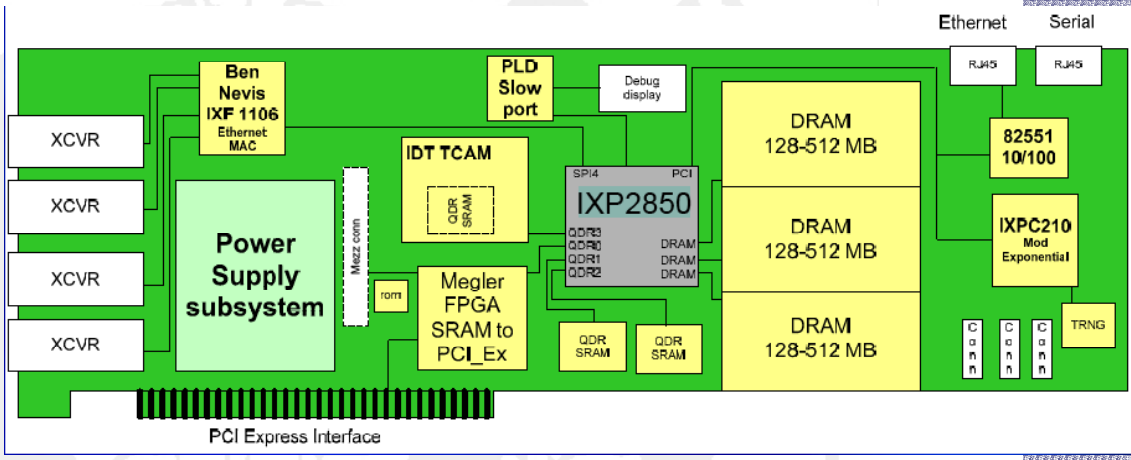


Recent funding: Smart NIC for HPC applications

- Address data movement in HPC domain
  - remote simulations, scientific visualization
  - HUGE data volumes -> important to tap into outputs and manage/prioritize their deliver in a application/user specific way
- Objectives
  - lightweight acceleration communication services and customization of their behavior
    - e.g., packet scheduling algorithm, protocol window rightsizing…
  - exploit OS- and application-bypass techniques
  - augment the communication specific services with application-specific codes
- DoE funding, interest by several national labs, joint with RNet
  - Greg Eisenhauer, Sudha Yalamanchili, Karsten Schwan, …

SmartNIC approach





- dotted line -> hardware accelerated layers
- Netronome platform permits access to entire host memory
  - currently evaluating `software' coherency
- already demonstrated feasibility of
  - using EVPath for application-specific tuning of service behavior
  - flexible DWCS scheduling
  - execution of additional EVPath handlers for additional content manipulation

Conclusions

- Importance of associating rich functionality near the network boundary
- Specialized hardware delivers numerous benefits
  - APIs should permit applications to take advantage of those
    - e.g., apply XML processing only to data of interest
    - e.g., increase flow's priority based on critical content
- Tight integration with main compute cores and memory important
  - however benefits from eliminating memory/IO loads important
- Tools!