Workload Analysis for Network Processor Design

Ramaswamy Ramaswamy, Ning Weng, Tilman Wolf

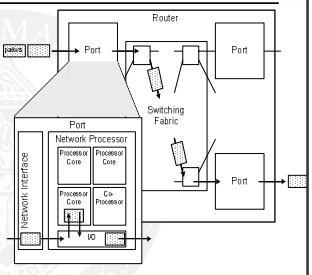
University of Massachusetts Amherst

Network Processing

- Computer Networks are becoming more versatile
  - Not just "store-and-forward"
  - More functionality performed "inside" network
    - Network Address Translation (NAT)
    - Firewalls
    - TCP/IP offloading
    - Virtual Private Network (VPN)
- Routers are equipped with port processors
  - "Network processors" (NPs)
- NPs are system-on-a-chip multiprocessors
  - Different from workstation/server processors
  - Simple, highly parallel workload
- System-level architecture of NPs is area of current research

Network Processors

- Router architecture
  - Ports connected through switching fabric
  - Processing is done on port
- Network Processors
  - Optimized for simple, I/O intensive tasks
  - Employ co-processors for address lookup, checksums
- What is best system architecture for NPs?
  - Processing resources, inter-connects, memory hierarchy
  - Depends on workload
- Our approach:
  - Workload analysis to understand application requirements
  - Analytic performance modeling to find optimal architecture



Outline

- Introduction
- Workload simulation
  - PacketBench tool
- Workload Analysis
  - Annotated DAG generation
  - Instruction clustering
- Task mapping for heterogeneous NP designs
- Current research problems
- Summary

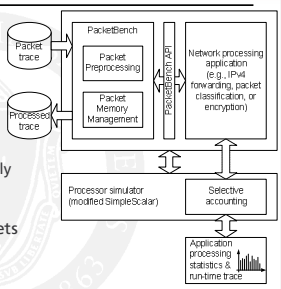
NP Workloads

- Useful to understand workload characteristics
  - Develop better network systems
  - Develop applications that benefit from network processing
  - Allocation of processing tasks to NP
- Network processing workload is unique
  - Different from workstation behavior
    - E.g., simplicity, high I/O
  - Dominated by small tasks
- Need to characterize processing behavior
  - Simulation
  - Need to simulate realistic packet processing environment
  - Separate application characteristics from framework

UMassAmherst

PacketBench

- PacketBench framework
  - Packet management
  - Simple API to application
- Simulation on SimpleScalar
  - ARM RISC processor typical for NP processing engines
  - Selective accounting considers only network processing application instructions
- Simulates processing or real packets with real applications
- Results:
  - Various application metrics (processing cost per packet, cache misses)
  - Run-time instruction trace



UMassAmherst

Workload Analysis

- Run-time instruction trace analysis
  - Better than static analysis (considers all dependencies)
  - Representative because of uniformity in packet processing
- Instructions and dependencies for directed graph
- Use clustering algorithm to group instructions
  - "Minimum Local Ratio-Cut"
- Resulting DAG is architecture-independent representation of application
  - Used for further analysis and mapping to processing resources

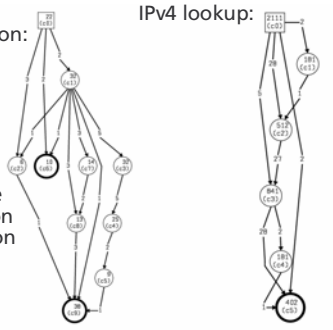
```
Inst.-#   Address     Instruction          Effective Address
...
129       33557096 : ldrb  r3,[r4,#8]     : 0x33977912
130       33557100 : cmp   r3,#0          : 0x--------
131       33557104 : bne   0x2000aa4      : 0x--------
132       33557156 : sub   r3,r3,#1       : 0x--------
133       33557160 : strb  r3,[r4,#8]     : 0x33977912
134       33557164 : mov   r2,#65280      : 0x--------
135       33557168 : ldr   r3,[r4,#8]     : 0x33977912
136       33557172 : add   r2,r2,#254     : 0x--------
137       33557176 : mov   r3,r3, lsr #16 : 0x--------
138       33557180 : cmp   r3,r2          : 0x--------
139       33557184 : mov   r2,#1          : 0x--------
...
```

UMassAmherst

Example Annotated DAGs

Flow classification:

IPv4 lookup:

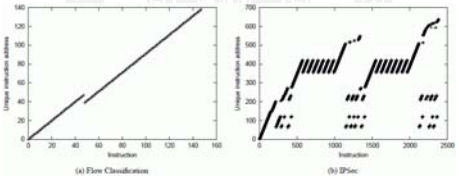


- ADAG size depends on termination criteria of clustering algorithm

UMassAmherst

Identification of Co-Processing

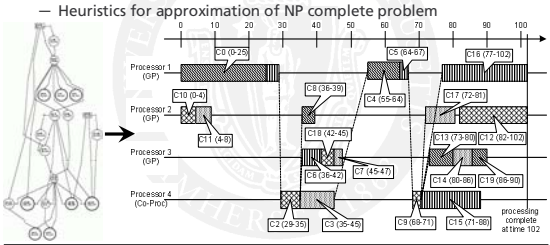
- NPs use co-processors for special tasks
  - Frequently used
  - Suitable for dedicated logic
- Application analysis can identify potential co-processing task
  - High reuse of same code (loops)



UMassAmherst

Task Mapping

- Mapping from ADAG to architecture yields schedule
  - Execution time of ADAG node depends on hardware resource
- Optimization criteria
  - Low delay or high throughput
  - Heuristics for approximation of NP complete problem



UMassAmherst

Current Research Questions

- Given application analysis and mapping algorithm. what is the optimal system architecture for an NP?
  - How do different applications impact architecture (e.g., parallel processors vs. pipelining)?
  - How does the optimal architecture depend on constraints (e.g., chip size, power consumption)?
- What programming abstractions are suitable for heterogeneous processing environments?
  - Programmers should be able to exploit
- How to model processing time for packets?
  - Processing causes considerable packet delays for complex applications
  - Needs to be considerable in network simulations

UMassAmherst

Summary

- Network processing is becoming increasingly important
- Workloads for NPs are very different from workstations/servers
  - Simple and highly parallel
- NP workload analysis
  - Annotated DAG
  - Instruction clustering
- DAG mapping to heterogeneous NP architectures
- Useful to address system-level NP design questions

UMassAmherst