Design Space Exploration of Network Processor Architectures

ECE 697J
December 3rd, 2002
Introduction

- Network processor architectures have many choices
  - Number of processors
  - Size of memory
  - Type of interconnect
  - Co-processors
- How can the optimal configuration be found?
  - Need to explore design space
- Simulations are too system-specific and limited
- Analytic performance modeling
  - Performance expressed as function of NP configuration and workload
System Components

• Model assumes SOC with:
  – Multiple processor cores
  – Micro-engines
  – Dedicated hardware (co-processors)
  – Memory units, caches
  – Interconnects
  – I/O interfaces

• Very general, matches most NPs
Analytic Model

- Performance model
  - Task (= processing requirements)
  - Resources (= NP resources)
  - Mapping of tasks to resources
  - Traffic (= arrival curves)
- Optimization (= cost function)
  - Chip area
  - On-chip memory
  - Performance
- Other properties
  - Delay
  - Throughput
Arrival Curves

**Definition 1 (Arrival Curves)** For any flow $f$, the lower arrival curve $\alpha^l_f$ and the upper arrival curve $\alpha^u_f$, satisfy the relation:

$$\alpha^l(t - s) \leq R(t) - R(s) \leq \alpha^u(t - s) \quad \forall 0 \leq s \leq t$$

$\alpha^l_f(\Delta)$ gives a lower bound on the number of packets that might arrive from a flow $f$ within any time interval of length $\Delta$. Likewise, $\alpha^u_f(\Delta)$ gives an upper bound on the number of packets that might arrive from a flow $f$ within any time interval of length $\Delta$. Hence, for all $\Delta > 0$, $\alpha^l_f(\Delta) \leq \alpha^u_f(\Delta)$ and $\alpha^l_f(0) = \alpha^u_f(0) = 0$. Therefore, within any time interval of length $\Delta \in \mathbb{R}_{\geq 0}$, the number of packets arriving from a flow $f$ is greater than or equal to $\alpha^l_f(\Delta)$, and less than or equal to $\alpha^u_f(\Delta)$. 
Arrival Curves

Figure 1. Representation of arrival curves.
Definition 2 (Task Structure) Let $F$ be a set of flows and $T$ be a set of tasks. To each flow $f \in F$ there is an associated directed acyclic graph $G(f) = (V(f), E(f))$ with task nodes $V(f) \subseteq T$ and edges $E(f)$. The tasks $t \in V(f)$ must be executed for each packet of flow $f$ while respecting the precedence relations in $E(f)$.
Task Structure

Figure 12. Task graph for a network process-
Deadlines and Requests

Definition 3 (Deadlines and Requests) To each flow $f \in F$ there is associated an end-to-end deadline $d_f$, denoting the maximum time by which any packet of this flow has to be processed after its arrival. If a task $t$ can be executed on a resource $s$, then it creates a “request”, denoting the processing requirement due to task $t$ processing a packet on the resource $s$. For example, this request might represent the number of processor cycles or instructions required for processing a packet with the function described by task $t$. Therefore, for all possible task to resource bindings there exist a request $w(t, s) \in \mathbb{R}_{\geq 0}$. 
System Architecture

Figure 2. Example of a physical (left) and logical (right) structure of a network processor architecture.
Definition 4 (Service Curves) For any $\Delta \in \mathbb{R}_{\geq 0}$ and any resource $s$ belonging to a set of available resources $S$, the lower service curve $\beta^l_s(\Delta)$ is a lower bound on the number of computing/communication units available from resource $s$ over any time interval of length $\Delta$. Similarly, the upper service curve $\beta^u_s(\Delta)$ denotes an upper bound on the number of computing/communication units available from resource $s$ over any time interval of length $\Delta$. Therefore, the computing/communication units available from resource $s$ over any time interval of length $\Delta$ is always greater than or equal to $\beta^l_s(\Delta)$ and less than or equal to $\beta^u_s(\Delta)$. 
Service Curves

Figure 3. Representation of service curves.
Definition 5 (Resources) We define a set of resource types $S$. To each type $s \in S$ there is associated a relative implementation cost $\text{cost}(s) \in \mathbb{R}_{\geq 0}$ and the number of available instances $\text{inst}(s) \in \mathbb{Z}_{\geq 0}$. To each resource instance there is associated a finite set of scheduling policies $\text{sched}(s)$ which the component supports, a lower service curve $\beta^l_s$ and an upper service curve $\beta^u_s$.

Definition 6 (Task to Resource Mapping) The mapping relation $M \subseteq T \times S$ defines possible mappings of tasks to resource types, i.e. if $(t, s) \in M$ then task $t$ could be executed on resource type $s$. 
Figure 13. Graphical representation of a part of the mapping of tasks to resources.
Analysis

- Packets processed independently
- Characteristic chain of tasks for each flow
- Develop scheduling network
- Apply real-time calculus from node to node
- Derive bounds
Scheduling Network
Arrival/Service Curve Transformation

\[ [\bar{\alpha}_l^i, \bar{\alpha}_u^i] \rightarrow [\beta_l^i, \beta_u^i] \rightarrow [\alpha_l^i, \alpha_u^i] \rightarrow [\alpha_l^i, \alpha_u^i] \rightarrow [\bar{\alpha}_l^i, \bar{\alpha}_u^i] \]

Eqn. (5) → Eqn. (1-4) → Eqn. (6)
Arrival/Service Curve Transformation

Theorem 1
Delay and Backlog

From: Robert Malaney & Glynn Rogers, CSIRO
Design Space Exploration

Figure 11. Basic concept for the design space exploration of packet processing systems.
Results
Results

Figure 15. Examples for Pareto-optimal resource allocations taken from Fig. 14. Darker coloring means higher average utilization.
Summary

- Analytic performance model for network processors
- Models traffic, processing, and interconnect
- Service curves give upper and lower bounds
  - Might diverge over many steps
- A bit difficult to apply
Alternative Model

- Use mean-value analysis
- Compute MIPS for overall system
- Determine cost function (e.g., area)
- Parameterize with realistic workload: CommBench
- Find optimal configuration / explore design space
NP System Model

- Single Chip Multi-processor
- Clusters:
  - Processors
  - Per-proc cache
  - Memory channel
- Processors are simple RISC cores
- Off-chip router functions:
  - Queuing
  - Packet demux
Design Parameters (1)

- Parameters that are considered in model:

<table>
<thead>
<tr>
<th>Component</th>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>processor</td>
<td>$clk_p$</td>
<td>processor clock frequency</td>
</tr>
<tr>
<td></td>
<td>$t$</td>
<td>number of simultaneous threads on processor</td>
</tr>
<tr>
<td></td>
<td>$\rho_p$</td>
<td>processor utilization</td>
</tr>
<tr>
<td>program a</td>
<td>$f_{load_a}$</td>
<td>frequency of load instructions</td>
</tr>
<tr>
<td></td>
<td>$f_{store_a}$</td>
<td>frequency of store instructions</td>
</tr>
<tr>
<td></td>
<td>$mi_{c,a}$</td>
<td>i-cache miss probability for cache size $c_i$</td>
</tr>
<tr>
<td></td>
<td>$md_{c,a}$</td>
<td>d-cache miss probability for cache size $c_d$</td>
</tr>
<tr>
<td></td>
<td>$dirty_{c,a}$</td>
<td>prob. of dirty bit set in d-cache of size $c_d$</td>
</tr>
<tr>
<td></td>
<td>$compl_{a}$</td>
<td>complexity (instr. per byte of packet)</td>
</tr>
<tr>
<td>caches</td>
<td>$c_i$</td>
<td>instruction cache size</td>
</tr>
<tr>
<td></td>
<td>$c_d$</td>
<td>data cache size</td>
</tr>
<tr>
<td></td>
<td>$linesize$</td>
<td>cache line size of i- and d-cache</td>
</tr>
<tr>
<td>off-chip memory</td>
<td>$\tau_{DRAM}$</td>
<td>access time of off-chip memory</td>
</tr>
</tbody>
</table>
Design Parameters (2)

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>memory channel</td>
<td>$width_{mch}$</td>
</tr>
<tr>
<td></td>
<td>$clk_{mch}$</td>
</tr>
<tr>
<td></td>
<td>$\rho_{mch}$</td>
</tr>
<tr>
<td>I/O channel</td>
<td>$width_{io}$</td>
</tr>
<tr>
<td></td>
<td>$clk_{io}$</td>
</tr>
<tr>
<td></td>
<td>$\rho_{io}$</td>
</tr>
<tr>
<td>cluster</td>
<td>$n$</td>
</tr>
<tr>
<td>ASIC</td>
<td>$m$</td>
</tr>
<tr>
<td></td>
<td>$s(x)$</td>
</tr>
</tbody>
</table>

- Develop performance model:
  1. Processor utilization
  2. Cache miss rate and memory access time
  3. Memory channel utilization
  4. Cluster configuration
### Processing Power

- **RISC:** one instruction every cycle unless stalled
- **Utilization** $\rho_p$ gives fraction of “useful” cycles
- **Total processing power:**

\[
IPS = \sum_{j=1}^{m} \sum_{k=1}^{n} \cdot \rho_{p_{j,k}} \cdot clk_{p_{j,k}}
\]

- If all processors are identical in configuration and workload:

\[
IPS = m \cdot n \cdot \rho_p \cdot clk_p
\]

- **Question:** How to determine $\rho_p$?
Processor Utilization

- Cache misses cause processor stalls
  - Reduce utilization
- Multithreading hides memory access latencies
- Processor utilization [Agarwal 1992]:

\[
\rho_p(t) = 1 - \frac{1}{\sum_{i=0}^{t} \left( \frac{1}{p_{\text{miss}} \cdot \tau_{\text{mem}}} \right)^i \frac{t!}{(t-i)!}}
\]

- Utilization decreases with
  - more cache misses \( p_{\text{miss}} \)
  - longer memory accesses \( \tau_{\text{mem}} \)
  - Fewer threads \( t \)
- Need to determine \( \tau_{\text{mem}} \) and \( p_{\text{miss}} \)
Memory System

• Memory access time has three components:
  – Queuing time until request is served
  – DRAM access time
  – Memory line transmission time

\[ \tau_{mem} = \tau_Q + \tau_{DRAM} + \tau_{transmit} \]

• DRAM access time fixed by technology used.
• Transmission time:

\[ \tau_{transmit} = \frac{\text{linesize}}{\text{width}_{mchl}} \cdot \frac{\text{clk}_p}{\text{clk}_{mchl}} \]

• Queuing time depends on load on memory channel.
Queuing Approximation

- Processors in cluster generate memory requests
  - Single server queuing system
  - Deterministic service time
  - Geometrically distributed inter-request time
- Approximation with waiting time in M/D/1 queue:

\[ \tau_Q = \frac{\rho_{mch}^2}{2(1 - \rho_{mch})} \cdot \frac{\text{linesize}}{\text{width}_{mch}} \cdot \frac{\text{clk}_{p}}{\text{clk}_{mch}} \]
On-Chip Caches

- Miss rate is combination of i-cache and d-cache misses:
  \[ p_{\text{miss},a} = m_{i,c,a} + (f_{\text{load}_a} + f_{\text{store}_a}) \cdot m_{d,c,a} \]

- Miss rates of application depend on effective cache size.
- Threads compete for cache \( \Rightarrow \) cache pollution
- Cache is effectively split among threads.
  - Effective cache size:
    \[ c_{i,\text{eff}} = \frac{c_i}{t}, \quad c_{d,\text{eff}} = \frac{c_d}{t} \]

- We now have expression for processor utilization.
Memory and I/O Channel

- How many processors can share one memory channel?
- Processor utilization and miss rates give memory bandwidth $bw_{mchl,1}$ of one processor.
- Number of processors that can share memory channel:

$$n = \left\lfloor \frac{width_{mchl} \cdot clk_{mchl} \cdot \rho_{mchl}}{bw_{mchl,1}} \right\rfloor$$

- Bandwidth for I/O channel depends on application:
  - Complex applications: little I/O
  - Simple applications: more I/O
  - Formal definition of “complexity” in paper
- Performance equation complete.
Chip Area

• Summation over all chip components:

\[
\text{area}_{NP} = s(io) + \sum_{j=1}^{m} (s(mchl) + \sum_{k=1}^{n} (s(p_{j,k}, t) + s(c_{i,j,k}) + s(c_{d_{j,k}})))
\]

• Processor size depends on number of thread contexts:

\[
s(p, t) = s(p_{basis}) + t \cdot s(p_{thread})
\]

• Memory channel size depends on channel width:

\[
s(mchl) = s(mchl_{basis}) + \text{width}_{mchl} \cdot s(mchl_{pin})
\]
Model Summary

• With IPS performance of system and chip area:
  – Compute IPS/area

• Necessary parameters:
  – Application parameters (load/store freq., cache miss rates, …)
  – Technology parameters (processor clock, component sizes, …)

• => Benchmark for application parameters
CommBench

- Network processor benchmark
- Benchmark applications:
  - Header-processing applications (HPA)
  - Payload-processing applications (PPA)

<table>
<thead>
<tr>
<th>HPA</th>
<th>PPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Deficit round robin</td>
<td>CAST encryption</td>
</tr>
<tr>
<td>IP header fragmentation</td>
<td>JPEG transcoding</td>
</tr>
<tr>
<td>Radix tree routing</td>
<td>Reed-Solomon FEC</td>
</tr>
<tr>
<td>TCP filtering</td>
<td>ZIP compression</td>
</tr>
</tbody>
</table>

- Two workloads:
  - A: HPA
  - B: PPA

- More details in [Wolf, Franklin 2000].
Application Parameters

- Workload characteristics for model evaluation
- Simple parameters
  - Can easily be measured
  - Easily adaptable to other workloads
Technology Parameters

- **0.18 µm CMOS technology**
- **Exact values are hard to get from industrial sources**
  - Performance model also works with more accurate parameters
- **Varied parameters:**
  - Processor clock
  - # of threads
  - Cache sizes
  - Memory channel bandwidth and load

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value(s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$clk_p$</td>
<td>200 MHz ... 800 MHz</td>
</tr>
<tr>
<td>$t$</td>
<td>1 ... 16</td>
</tr>
<tr>
<td>$c_i$</td>
<td>1 kB ... 1024 kB</td>
</tr>
<tr>
<td>$c_d$</td>
<td>1 kB ... 1024 kB</td>
</tr>
<tr>
<td>linesize</td>
<td>32 byte</td>
</tr>
<tr>
<td>$\tau_{DRAM}$</td>
<td>60 ns</td>
</tr>
<tr>
<td>width_mchl</td>
<td>16 bit ... 64 bit</td>
</tr>
<tr>
<td>$\rho_{mch}$</td>
<td>0 ... 1</td>
</tr>
<tr>
<td>width_io</td>
<td>up to 72 bit</td>
</tr>
<tr>
<td>$\rho_{io}$</td>
<td>0.75</td>
</tr>
<tr>
<td>$clk_{mch}, clk_{io}$</td>
<td>200 MHz</td>
</tr>
<tr>
<td>$s(p_{basis})$</td>
<td>1 mm$^2$</td>
</tr>
<tr>
<td>$s(p_{thread})$</td>
<td>0.25 mm$^2$</td>
</tr>
<tr>
<td>$s(c_i), s(c_d)$</td>
<td>0.10 mm$^2$ per kB</td>
</tr>
<tr>
<td>$s(mch_{basis}), s(io_{basis})$</td>
<td>10 mm$^2$</td>
</tr>
<tr>
<td>$s(mch_{pin}), s(io_{pin})$</td>
<td>0.25 mm$^2$</td>
</tr>
<tr>
<td>$s(ASIC)$</td>
<td>up to 400 mm$^2$</td>
</tr>
</tbody>
</table>
Results

• Optimal configurations
• Performance trends (take optimal configuration and vary parameter)
  – Memory channel
  – Processor clock and threads
  – Caches
• Note: performance metric is MIPS/mm²
Optimal Configuration

- **Processor:**
  - 800 MHz
  - 2 threads
  - ~96% utilization

- **Chip configuration:**
  - 2-3 clusters with 20-30 processors
  - 16-32 kB caches (instruction and data)

- **Memory channel:**
  - ~90% load
  - 64 bit width

- **Off-chip memory:**
  - 120-140 cycles access time

- **Area:**
  - 140-180 mm²

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Workload A</th>
<th>Workload B</th>
</tr>
</thead>
<tbody>
<tr>
<td>$clk_p$</td>
<td>800 MHz</td>
<td>800 MHz</td>
</tr>
<tr>
<td>$t$</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>$m$</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>$c_i$</td>
<td>16 kB</td>
<td>32 kB</td>
</tr>
<tr>
<td>$c_d$</td>
<td>16 kB</td>
<td>16 kB</td>
</tr>
<tr>
<td>$width_{mch}$</td>
<td>64 bit</td>
<td>64 bit</td>
</tr>
<tr>
<td>$\rho_{mch}$</td>
<td>0.91</td>
<td>0.89</td>
</tr>
<tr>
<td>$p_{miss}$</td>
<td>0.187%</td>
<td>0.286%</td>
</tr>
<tr>
<td>$\tau_{mem}$</td>
<td>137.6</td>
<td>121.6</td>
</tr>
<tr>
<td>$\rho_p$</td>
<td>0.974</td>
<td>0.957</td>
</tr>
<tr>
<td>$n$</td>
<td>31</td>
<td>20</td>
</tr>
<tr>
<td>$width_{io}$</td>
<td>71</td>
<td>3</td>
</tr>
<tr>
<td>$pins_{NP}$</td>
<td>$199+pins_{control}$</td>
<td>$195+pins_{control}$</td>
</tr>
<tr>
<td>IPS</td>
<td>48324 MIPS</td>
<td>45934 MIPS</td>
</tr>
<tr>
<td>area</td>
<td>272 mm²</td>
<td>322 mm²</td>
</tr>
<tr>
<td>IPS/area</td>
<td>178 MIPS/mm²</td>
<td>142 MIPS/mm²</td>
</tr>
</tbody>
</table>
Memory Channel

- Best load ~90%
- Low load:
  - Waste of area for memory channel
- High load:
  - Very long queue length in M/D/1 model
  - High memory access time
Practically linear growth with processor clock speed

- Less significant with more threads
  - Less cache per thread
  - Performance limited by off-chip memory access time
Cache Configuration

- Cache size
  - Small caches cause inefficient execution
  - Large caches waste space

- Performance very sensitive to deviations from optimum
Summary

• NP performance model
  – Determines processing performance of NP configuration
  – Relates processing power to area of system-on-a-chip
  – Uses simple workload characteristics and technology parameters

• Optimal configuration for given scenario

• Performance trends as “rules of thumb:”
  – Cache configuration has big impact on performance
  – Two to four thread contexts is optimal
  – Higher processor clock rates and memory channel directly translate into higher performance

• Model can aid in first-order NP design
Projects

- Due: 12/10/02
- Turn in:
  - Report (about 10 pages)
  - Basically conference paper style
- In class:
  - 10-12 minute presentation
  - Brief repetition of problem
  - Focus on results
Final

- December 16\textsuperscript{th}, 10:30am-12:00pm, MRST 220
- Comprehensive
- Similar to midterm