Networking Applications

Janani Ravi

Metha Jeeradit

John Kim

Outline

- Introduction
- Techniques to improve performance
  - Caching
  - Parallelism
- Example: media streaming
- Network processors
- Benchmarks

Introduction

- Network can be considered as a distributed information system providing access to shared data
- Due to rapid growth of Internet, networking applications usually span large geographical area raising several issues e.g.,
  - Network latency tolerance
  - Error and loss handling
  - Delay requirements for real-time apps

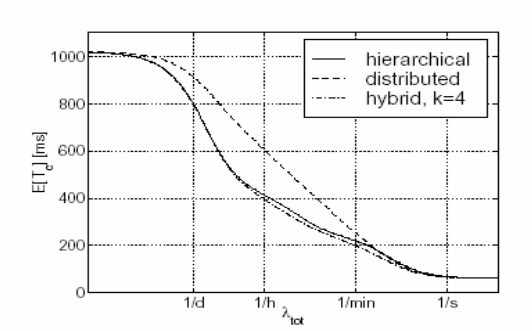
Web Caching

- Proxy servers can be used to cache recently used documents
  - Reduces bandwidth consumption, access latencies, workload of remote servers and improves availability of cached documents
  - Possible access to stale data, extra processing due to cache miss, single proxy would be a bottleneck

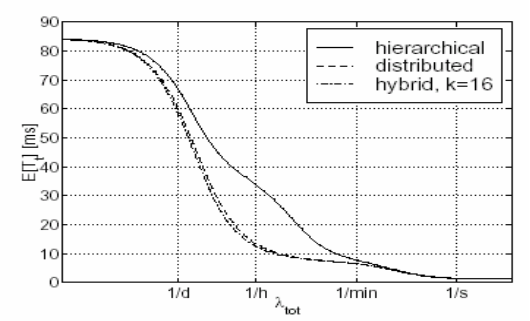
Caching Architectures

- Hierarchical
  - Shorter access times but…
  - Every level adds delay, higher level caches bottlenecks, multiple copies at different levels
- Distributed
  - Shorter transmission times and better bandwidth usage but…
  - Higher bandwidth usage overall
- Hybrid

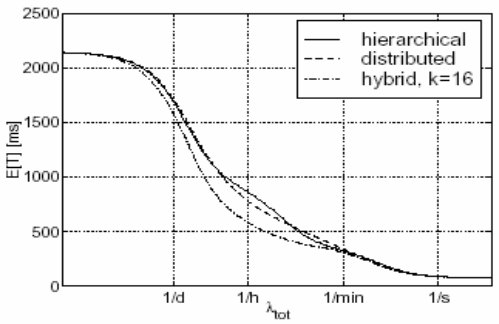
Latencies



Connection time



Transmission time



Total latency

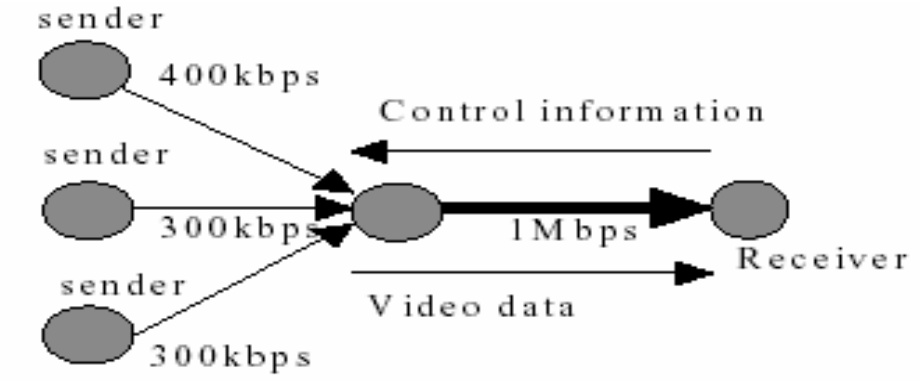
Prefetching

- Local
  - Based on local information like reference patterns
- Server-hint
  - Based on content specific knowledge of objects and reference patterns of a far greater number of clients

Parallelism

- Layer level
- Connection level
- Functional level
- Packet Level

Example: Media Streaming



(multiple senders, one receiver)

Requirements:

- Ability to tolerate errors and jitters
- High network bandwidth
- Delay and performance guarantees

Example: Media Streaming

Receiver side:

- Estimate sender's BW at a regular interval and perform rate allocation
- TLP in BW estimation since each sender's BW can be estimated independently
- Sequential computation in rate allocation

Example: Media Streaming

Sender side:

- Keep track of difference between est. arrival time and playback time of each packet
- If max difference for current sender, then send packet
- TLP and PLP in difference estimation
- PLP in sending the packets

Example: Media Streaming

Computational Complexity

- Roughly constant with increasing number of senders with enough hardware support

Performance Bottleneck:

- Inter-process communication between the application and the operating system
- High context switching cost (e.g., on Windows NT)

Architecture : GPU or NPU?

- History of processors used in networking
  - software-based – flexibility
  - ASIC - performance
  - specialized network processors
    - Programmability
    - Performance
    - Management
    - Routing

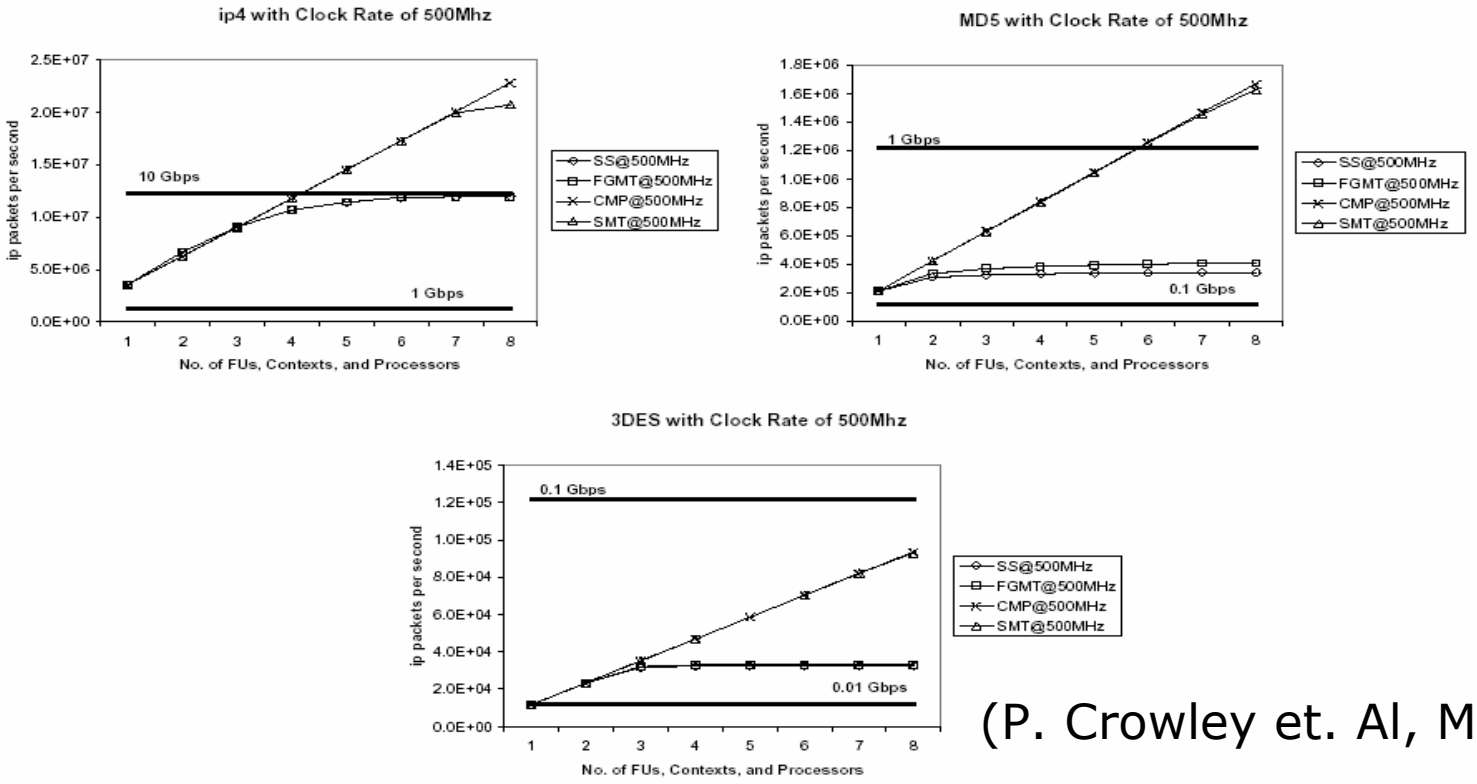
Architecture : GPU or NPU?

- Moore's Law vs Network requirement
- Performance : avg. case vs. worst case
  - Locality & speculation
- Real-time requirement
- Power budget
- ILP vs TLP (packet-level parallelism)
- Multithreaded architecture

Network Processors

- Specialized chips to handle data trafficking
- 4 network processors evaluated
  - Superscalar
  - Fine-grained multithreaded
  - Chip-multiprocessor
  - Simultaneous multithreaded

Performance Comparison



(P. Crowley et. Al, May '00)

Figure 7. Performance results for all architectures, clocked at 500 MHz, running all benchmarks.

Benchmarks Available

- EEMBC – industry standard benchmark in embedded application
- MiBench – cheap version of EEMBC
- Mediabench – multimedia & communication
- Netbench – target network processors
- Commbench – packet based processing tasks such as routing&data forwarding

Benchmarks

- Other benchmarks

Application	Insts Executed per Message	Loads/Stores (%)	Ctrl Flow (%)	Other (%)
IP forward	~200	25.4	12.7	61.9
MD5	~2000	10.7	2.8	86.5
3DES	~40000	17.8	1.2	81.0