CS228A: Architectural and OS Models

George Varghese

January 15, 2002
Architectural Models

- Optimizing network performance requires optimizing the path of data through the internals of the source node, sink node and every router.
- Thus it is important to understand the internal architecture of endnodes and routers.
  - They are different: endnode architectures optimized for general computation, and router architectures for Internet communication.
Alternative Endnode Architecture?

- Packets from network can be placed in Memory 2 while Processor reads from Memory 1. Can alternate between memories if Processor wishes to read Memory 2.

- Some proposals (e.g., Infiniband as a replace for the PCI bus).
Network Processors

- General purpose programmable processors optimized for network traffic motivated by unpredictable router tasks and high cost of NRE like masks.

- Often use many processors that work on different packets at the same time. Intel IXP1200 has 6 177 MHz processors with 5.6 nsec clock cycle.

- Often use multithreading to overcome memory latencies. In IXP, 32 bytes to DRAM (read a header) takes 45 clocks, forwarding rate of 900K packets per processor. To get around this, each processor works on multiple packets, when one packet stalls, processor works on another. Requires fast context switching (IXP has 4 contexts).

- Optionally have special purpose instructions for address lookup. Header gluing.
Buffering and Optical Switching

- Optics is great. Avoids electrical bottleneck. Can even switch light with mirrors and prisms (Lithium Niobate devices). So why not build an all-optical router.

- Achilles heel: buffering. So it’s easy to build and optical circuit switch with electronic control. Used by providers to change bandwidth provisioning between POPs at a coarse granularity (Litera). Hot area.
Operating Systems

None on routers. But need to understand OS to improve packet forwarding over endnodes. Need to understand costs of abstractions (idealizations we invent to deal with the perversity of the real world.)


- *Infinite Memory:* Illusion of infinite, contiguous and unshared memory.

- *Simple I/O:* Avoid dealing with dubious details of devices, simpler illusion of reading and writing devices from memory.
Uninterrupted Computation via Processes

- 3 underlying mechanisms: Context switching (save and restore), scheduling, protection
- 3 flavors of “process”: interrupt handlers, threads, and user processes - order of increasing generality and cost.
Example: Receiver Liveloop in BSD Unix

- Receiver liveloop: keep processing incoming packets only to discard them later because applications never run.

- Latency depends on process, thread times. On Pentium II, interrupt (10 usec), 10-100 usec (Context switch on Linux machine). Large.
Page mapping (avoids finding contiguous locations) and demand paging (lets us use more space than main memory).

Slow DRAM lookup of page table avoided by using a cache (TLB) done by MMU.

Protection by allowing only OS to modify page tables and using read/write protection bits
Simple I/O using System Calls

- If abstraction was only concern, I/O calls could be libraries. For security, I/O calls handled by device drivers in a core of essential services (kernel) that applications cannot be trusted to do.

- System calls, trap to kernel protection levels. More expensive function call because of hardware privilege escalation, and sanitizing checks. 2.31 usec on Pentium II 100 using Linux. Affects networking.
A Manifesto for Change

There’s another aspect of creativity. We’ve been talking about great individual contributors, but when you’re creating technology it’s necessarily a group process, because technology today is so complex that it has to be interdisciplinary. . . And they’re all essentially speaking their own languages. So we will spend months establishing our common language . . . I have a technique to get people to think outside the box: I’ll give a signal-processing problem to the linguists, and vice versa, and let them apply the disciplines in which they’ve grown up to a completely different problem. The result is often an approach that the experts in the original field would never have thought of. Group process gives creativity a new dimension. — R. Kurzweil