• Application papers → talk about different parallel programming models

Supra-linear Packet Processing Performance with Intel Multi-Core Processors:

• Snort: process TCP packets for intrusion detection. Not a firewall (would do much deeper checking).
  ○ Flow-level stateful. Signature-based (hash of some portion of packet), compares to signature list it has
  ○ Operates on raw packets, not on sockets

Networking basics

• Ethernet uses frames
  ○ IP packets. Use Maximum Transmission Unit (MTU) discovery to figure out how large IP packets should be. If larger than MTU, IP packets get fragmented. Many places drop IP fragments, because common vector for DoS
    ■ TCP flows: flow-controlled, long-lived, bidirectional.
- Snort uses flow pinning
- UNIX API: socket abstraction. Underlying layers responsible for reassembly
  - For performance, Snort processes packets. Sorts into flows, so it can look for fragmented malware
- Snort keeps state per flow
  - If packets go round robin, may need to move flow state around
- Figure 3: Single core; no parallelism
- Figure 4: All cores executing same program in parallel
  - Work distributed round-robin
  - Cores need to communicate through memory/cache coherence to track state on a per-flow basis
- Figure 5: Core 1 allocates packets to different cores based on hash of src/dest IP address/ports
  - 4-tuple hash is unique enough to ID flows; unique tuple since src port often changes when making multiple TCP connections from same machine to same destination
  - This keeps the state of each flow local to a particular core
  - Snort hashing can lead to performance imbalance; load imbalance can occur when there is an elephant flow
- Supra linear: increase cache size effectively, can keep working set in cache
- Follows StreamIt requirements for streams

Performance of Database Workloads on Shared-Memory Systems with Out-of-Order Processors

- Authors
  - Sarita Adve → UIUC
  - Partha Ranganathan → Platforms group at Google, previously HP research
  - Luiz Barroso: Was @ DEC (Alpha, Vax), now @ Google. Ran platforms division
- Two applications: online transaction processing (OLTP), decision support systems (DSS)
  - OLTP: transactions, ecommerce, banking
  - DSS: big data, data mining
- Simulation using RSIM and traces
  - Trace: of instructions being run
  - Trace gathered from Alpha ISA machine, 4 processor system
- One of first multiprocessor exploration papers

- Traces for simulation of multicore?
  - Microarchitecture changes will change order of execution of instructions → matters for synchronization (e.g. lock acquisition order might change). Result in real microarchitecture seeping into performance model
• OLTP/DSS
  - Building bigger, beefier processors can provide some performance improvement
  - DSS: optimizations pretty effective
  - OLTP: optimizations don't provide as much benefit as DSS
  - Contrast with argument of Clearing the Clouds paper (to be discussed next class)