Module 2: "Parallel Computer Architecture: Today and Tomorrow"

Lecture 4: "Shared Memory Multiprocessors"

The Lecture Contains:

[From Chapter 1 of Culler, Singh, Gupta]

Previous Next

Technology trends

- The natural building block for multiprocessors is microprocessor
- Microprocessor performance increases 50% every year
- Transistor count doubles every 18 months
  - Intel Pentium 4 EE 3.4 GHz has 178 M transistors on a 237 mm2 die
  - 130 nm Itanium 2 has 410 M transistors on a 374 mm2 die
  - 90 nm Intel Montecito has 1.7 B transistors on a 596 mm2 die
- Die area is also growing
  - Intel Prescott had 125 M transistors on a 112 mm2 die
- Ever-shrinking process technology
  - Shorter gate length of transistors
  - Can afford to sweep electrons through channel faster
  - Transistors can be clocked at faster rate
  - Transistors also get smaller
  - Can afford to pack more on the die
  - And die size is also increasing
  - What to do with so many transistors?
- Could increase L2 or L3 cache size
  - Does not help much beyond a certain point
  - Burns more power
- Could improve microarchitecture
  - Better branch predictor or novel designs to improve instruction-level parallelism (ILP)
- If cannot improve single-thread performance have to look for thread-level parallelism (TLP)
  - Multiple cores on the die (chip multiprocessors): IBM POWER4, POWER5, Intel Montecito, Intel Pentium 4, AMD Opteron, Sun UltraSPARC IV
- TLP on chip
  - Instead of putting multiple cores could put extra resources and logic to run multiple threads simultaneously (simultaneous multi-threading): Alpha 21464 (cancelled), Intel Pentium 4, IBM POWER5, Intel Montecito
- Today's microprocessors are small-scale multiprocessors (dual-core, 2-way SMT)
- Tomorrow's microprocessors will be larger-scale multiprocessors or highly multi-threaded
  - Sun Niagara is an 8-core (each 4-way threaded) chip: 32 threads on a single chip

Architectural trends

- Circuits: bit-level parallelism
  - Started with 4 bits (Intel 4004) [http://www.intel4004.com/]
  - Now 32-bit processor is the norm
  - 64-bit processors are taking over (AMD Opteron, Intel Itanium, Pentium 4 family); started with Alpha, MIPS, Sun families
- Architecture: instruction-level parallelism (ILP)
  - Extract independent instruction stream
  - Key to advanced microprocessor design
  - Gradually hitting a limit: memory wall
  - Memory operations are bottleneck
  - Need memory-level parallelism (MLP)

- Also technology limits such as wire delay are pushing for a more distributed control rather than the centralized control in today's processors

- If cannot boost ILP what can be done?
- Thread-level parallelism (TLP)
  - Explicit parallel programs already have TLP (inherent)
  - Sequential programs that are hard to parallelize or ILP-limited can be speculatively parallelized in hardware
    - Thread-level speculation (TLS)
- Today's trend: if cannot do anything to boost single-thread performance invest transistors and resources to exploit TLP

Previous Next

Exploiting TLP: NOW

- Simplest solution: take the commodity boxes, connect them over gigabit ethernet and let them talk via messages
  - The simplest possible message-passing machine
  - Also known as Network of Workstations (NOW)
  - Normally PVM (Parallel Virtual Machine) or MPI (Message Passing Interface) is used for programming
  - Each processor sees only local memory
  - Any remote data access must happen through explicit messages (send/recv calls trapping into kernel)
- Optimizations in the messaging layer are possible (user level messages, active messages)

Supercomputers

- Historically used for scientific computing
- Initially used vector processors
- But uniprocessor performance gap of vector processors and microprocessors is narrowing down
  - Microprocessors now have heavily pipelined floating-point units, large on-chip caches, modern techniques to extract ILP
- Microprocessor based supercomputers come in large-scale: 100 to 1000 (called massively parallel processors or MPPs)
- However, vector processor based supercomputers are much smaller scale due to cost disadvantage
  - Cray finally decided to use Alpha µP in T3D

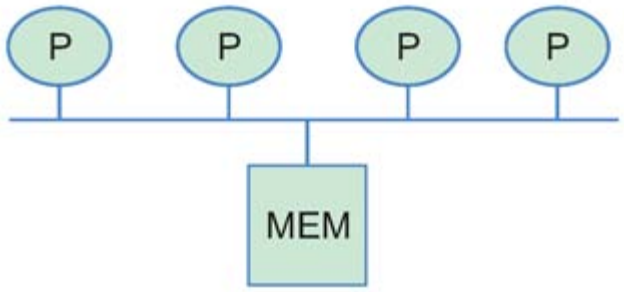
Exploiting TLP: Shared memory

- Hard to build, but offers better programmability compared to message-passing clusters
- The “conventional” load/store architecture continues to work
- Communication takes place through load/store instructions
- Central to design: a cache coherence protocol
  - Handling data coherency among different caches
- Special care needed for synchronization

Previous Next

Shared memory MPs

- What is the communication protocol?
  - Could be bus-based
  - Processors share a bus and snoop every transaction on the bus



  - The most common design in server and enterprise market

Bus-based MPs

- The memory is “equidistant” from all processors
  - Normally called symmetric multiprocessors (SMPs)
- Fast processors can easily saturate the bus
  - Bus bandwidth becomes a scalability bottleneck
  - In `90s when processors were slow 32P SMPs could be seen
  - Now mostly Sun pushes for large-scale SMPs with advanced bus architecture/technology
  - The bus speed and width have also increased dramatically: Intel Pentium 4 boxes normally come with 400 MHz front-side bus, Xeons have 533 MHz or 800 MHz FSB, PowerPC G5 can clock the bus up to 1.25 GHz

Scaling: DSMs

- Large-scale shared memory MPs are normally built over a scalable switch-based network
- Now each node has its local memory
- Access to remote memory happens through load/store, but may take longer
  - Non-Uniform Memory Access (NUMA)
  - Distributed Shared Memory (DSM)
- The underlying coherence protocol is quite different compared to a bus-based SMP
- Need specialized memory controller to handle coherence requests and a router to connect to the network

Previous Next

On-chip TLP

- Current trend:
  - Tight integration
  - Minimize communication latency (data communication is the bottleneck)
- Since we have transistors
  - Put multiple cores on chip (Chip multiprocessing)
  - They can communicate via either a shared bus or switch-based fabric on-chip (can be custom designed and clocked faster)
  - Or put support for multiple threads without replicating cores (Simultaneous multi-threading)
  - Both choices provide a good cost/performance trade-off

Economics

- Ultimately who controls what gets built?
- It is cost vs. performance trade-off
- Given a time budget (to market) and a revenue projection, how much performance can be afforded
- Normal trend is to use commodity microprocessors as building blocks unless there is a very good reason
  - Reuse existing technology as much as possible
- Large-scale scientific computing mostly exploits message-passing machines (easy to build, less costly); even google uses same kind of architecture [use commodity parts]
- Small to medium-scale shared memory multiprocessors are needed in the commercial market (databases)
- Although large-scale DSMs (256 or 512 nodes) are built by SGI, demand is less

Summary

- Parallel architectures will be ubiquitous soon
  - Even on desktop (already we have SMT/HT, multi-core)
  - Economically attractive: can build with COTS (commodity-off-the-shelf) parts
  - Enormous application demand (scientific as well as commercial)
  - More attractive today with positive technology and architectural trends
  - Wide range of parallel architectures: SMP servers, DSMs, large clusters, CMP, SMT, CMT, …
  - Today's microprocessors are, in fact, complex parallel machines trying to extract ILP as well as TLP

Previous Next