CSE 506: Operating Systems

MP Scheduling

Symmetric Multi-Processing (SMP)

• All CPUs similar, equally “close” to memory
• Horribly abused name by software community
  – Use “SMP” for anything with more than 1 “context”

Multi-core (CMP)

• All CPUs inside a single chip

Non-Uniform Memory Access (NUMA)

• Want to keep execution near memory
  – Accessing “remote” memory is more expensive

Hyper-Threading (SMT)

• One core, but multiple contexts
  – What’s a context?
    • A set of register values (including ones like CR3)
• OS view: 2 logical CPUs
  – “CPU” is also horribly abused
  – Really should be “hardware context” or “hardware thread”
  – Does not duplicate execution resources
  – Programs on same core may interfere with each other
    • But both may run
      – 2x slow threads may be better than 1x fast one

• All CPUs inside a single chip
All Kinds of Parallelism Together

- 2-socket NUMA, w/2 dual-threaded cores per socket

One set of Run Queues per “CPU”

Rebalancing Tasks
- Once task in one CPU’s runqueue
  - It stays on that CPU?
- What if all processes on CPU 0 exit
  - But all of the processes on CPU 1 fork more children?
- We need to periodically rebalance
  - CPU that runs out of work does the rebalance
  - work stealing
- Balance overheads against benefits
  - Figuring out where to move tasks isn’t free

Scheduling Domains
- General abstraction for CPU topology
- “Tree” of CPUs
  - Each leaf node contains a group of “close” CPUs
- When a CPU is idle, it triggers rebalance
  - Most rebalancing within the leaf
  - Higher threshold to rebalance across a parent
- What if all CPUs are busy
  - But some have fewer running tasks than others?
  - Might still want to rebalance
    - Heuristics in scheduler to decide when to trigger rebalance

NUMA Scheduling Domains

- Higher threshold to move to sibling/parent
NUMA + Hyperthreading

Rebalancing Strategy

- Read the loadavg of each CPU
  - Find the one with the highest loadavg
- Figure out how many tasks we should take
  - If worth it, take tasks
    - Need to lock runqueue
  - If not, try again later