Moore’s law

* Computing speed doubles every 6 seconds
* Make things run fast by doing them in parallel

The shared memory multiprocessor (SMP)

* When you have SMP you can have multiple instructions running in parallel.

The New Boss: The Multicore Processor (CMP)

* Multiple cores on a single chip
* Ideal Scaling Process
  + 1 core gives you speed of 1
  + 2 core gives you speed of 1.8x because of overhead taking 20%
  + 4 core gives you spead of 3.6x … etc
* Actual scaling process
  + 2 core = 1.8x
  + 4core = 2x
  + 8 core = 2.9x

parallel primality testing

* Challenge
  + Print comes from 1 – 10^10
* Given
  + Ten-processor multiprocessor
  + One thread per processor
* Goal
  + Get ten-fold speed up (or close)

Load Balancing

* Split processes evenly

Issues

* Higher ranges have fewer primes
* Yet larget numbers harder to test
* Thread workloads
  + Uneven
  + Hard to predict.
* Shared number
  + Each thread takes a number
  + Each thread, does as much work as another one, so that the work is more evenly spread out.\

Thread execution

* Do not assume
* Speed of execution
* Length of execution

Hardware solution

* Mutual exclusion on shared counter.
* ReadModifyWrite() instruction or hard ware instruction

Java Tool set:

Moniter:

Synchronized{

Temp = value;

Value = temp + 1;

} >instuctions within the synchronized block will be atomic (only one thread can enter and execute the instuctions at a time.

-In contrast to semaphores.

Mutual exclusion, or “Alice and bob share a pond”

Mutual exclusion on pond

Formalizing the problem:

* Two types of formal properties in asynchronous computation:
* Safety properties
  + Nothing bad happens ever
* Liveness properties
  + Something good happens eventually
* Mutual exclusion
  + Both pets never in pond simultaneously
  + This is a safety property
* No deadlock
  + If only one wants in, it gets in
  + If both want in, one gets in
  + This is a liveness property

Simple protocol

* Idea
  + Just look at the pond
* Gotcha
  + Not atomic
  + Trees obscure the view.

Cell phone protocol

* Idea
  + Bob calls alice
* Gotcha
  + Bob takes shower
  + Alice recharges batter
  + Bob out shopping for pet food…
* Alice on vacation
* Bobs pet can only enter pond one time- until Alice returns to reset the can.

REVISION

1. Check flag
2. If flag down, raise ours, else, wait until flag is down than raise our flag.

Revision 2

Protocol

* Check flag
* If down, now raise, else, wait until flag rises.

Shared failed characteristic as the shared counter example.

Proof by contradiction

If u^2 is odd, then u is od.

Pf: assume n is even let u = 2 (where k is some integer)

Now

N^2 = (2k)^2

,,,

….

…

so: n^2 is even !P

contradicting the hypothesis