IN3030/IN4330 Lecture
2022-05-19

Lecture will start at 10:15
Today

Agenda:

- **Review of course**
- **Review of old exams**

- **Threads in Java**

- **Concurrency:**
  - **Cyclic barrier**
  - **Semaphores**
- **Parallelization of sequential algorithms**
  - **Divide and conquer**
  - **Recursion – and thread**
- **Caching**
  - **Slow speed of light: the latency cause by it**

```
T0: s/b …    A0            s2/b    B0        c
T1:    s/r0  A1                  s2/r0 B1          c
T2:                 s/b   A2           s2/b  B2 c
```

T3: s/r2 A3 s2/r2 B3 c

Printed: A1 A0 A3 A2 B1 B0 B3 B2

T0: s/b A0 s2/r3 B0 c
T1: s/b
T2: s/r0 A2 s2/b B2 c
T3: s/b A3 s2/r2 B3 c

Printed: A0 A2 B0 A3 B3 B2

T0: s/b A0 s2/r2 B0 c
T1: s/r0 A1 s2/b B1 c
T2: s/r3 A2 s2/b B2 c
T3: s/b A3 s2/r2 B3 c

Printed: A1 A0 B1 B0 A2 A3 B3 B2

Lecture restarts at 11:15

Parallel Bubblesort:

Locks L2 L2 U L1 L1 L0 L0

Indexes: 0 1 2 3 4 5 6

T0

T1 R

T2

T3 B2

Instead of locking just ONE element, lock K element where K for example is 10,000

IF L cores, then choose for example 2*L threads, K = N/8*(2*L)/10,000

Review

Main points:

1. Emphasis on performance of code – effective and efficient implementation of algorithms.
2. Caching: the effect of caching can be significant.

3. Exploiting parallelism: modern computers have CPUs with multiple cores and can do lots of work in parallel – keep all cores working on your problem
4. Cutting up the work:
    a. divide the data
    b. divide the work
5. Methods:
    a. Converting a recursive algorithm to a parallel algorithm.
    b. Threadpool/taskpools
6. Techniques: Synchronization
    a. when threads have to access/write shared data, they must synchronize
    b. Monitors
    c. Cyclic barrier
    d. Atomic Integer
    e. Semaphores
    f. Java's "synchronized" keyword

7. Tradeoff between programmer's work vs the computer working for you
    a. Amdahl's law – Gufstavson's
8. Driving forces behind parallel programming:
    a. the slooow speed of light: merely 300,000 km/s: 3 GHz CPU: 1 cycle about 10 cm
    b. in copper cables: 180,000 km/s
    c. in fiber: 220,000 km/s
    d. Moore's "law"/observation: doubling of transistor every 1-2 years
    e. hard limit on CPU speed caused by quantum mechanical effects.
    f. After 2005: number of cores that double every 1-2 years – until about 2018
9. The Art of Parallel Programming – guessing supported by measurements

Skill tests:

How to produce a parallel version of a sequential algorithm?
One way is to find a parallel solution that solved the given problem.
Find a parallel version of an algorithm where the parallel version is *true to the spirit of the original*.
Testing synchronization: using a problem example.