Exam 24.11.2016

**Question 1: Concurrent linked list**

**1.a**

Simplest, coarse grained, implementation is to just lock entire list whenever mutating and reading it: (presented in pseudocode)

Let globalLock = lock

add(int value) {

lock.acquire()

int key = find suitable key…

list.add(key, value)

lock.release()

}

remove(int key) {

lock.acquire()

list.remove(key)

lock.release()

}

**1.b**

A problem in concurrent programming where a process is continuously denied access.

Starvation should not be possible as the solution in a) basically functions as if it was a single threaded program. However if an insertions or removal somehow fails, and it is not handled, possibly the lock will never be released resulting in a deadlock, or starvation. Or if the system have a bad scheduler.

**1.c**

Contention will happen whenever two or more threads are asked to do either a remove or insert, within the timespan of one remove/insert operation, as the last thread will have to wait for the first to release the lock. This could be improved by only locking the specific element of interest.

Since each thread locks entire table, contention will always be present as long as more than 1 request is being handled. Thus contention will always delay the threads.

The course has not provided me with any way to estimate how long time a given process takes to execute, but say mutations all take 2 units of time, and read operations take 1 unit of time, then read operations will only give half the contention as two of them can execute in the time one mutation takes.

**1.e)**

If mutation require 3 nodes with a lock on each, and the add takes lock 2 and 3, remove takes lock 0 and 1. Add needs lock 1 to execute and remove needs lock 2 to execute resulting in deadlock.

**Question 3: Cloud**

**3.a)**

A form of distributed computing, where computers and servers are accessed via a network, data and computation does not need to happen on the client machine, but can happen on the distributed remote network (cloud).

**3.b)**

IaaS: Infrastructure as a service. Mostly data storage and servers. E.g. dropbox

PaaS: Platform as a service. Platform in the cloud, developers can run applications on. This provide access to tools and stuff. Don’t need to worry about OS and updates for example

SaaS: Software as a Service. Basically everything is run in the cloud e.g. software. Client need not install stuff, they can just access through browser.

**3.c)**

Microsoft Azure provides SaaS, which means they also provide PaaS and IaaS.

**3.d)**

Assuming each part is run one after another, assembly provide bottleneck for DRAM, which means only 3 can compute at a time. Therefore we can only utilize 3 cores, thus computation time is (6 h + 1/60 h + 2048 h) / 3