Universität Heidelberg
Institut für Technische Informatik

Prof. Dr. Robert Strzodka
Steffen Lammel

Advanced Parallel Computing
Term 2015 (Summer)

Exercise 4

- **Return electronically (MOODLE) until Tuesday, 19.05.2015 14:00**
- **Include name on the top sheet. Stitch several sheets together.**
- **A maximum of two students is allowed to work jointly on the exercises.**

4.1Reading

Read the following papers and provide reviews as explained in the first lecture (see slides):

- A. Kägi, D. Burger, J. Goodman, Efficient Synchronization: Let Them Eat QOLB, ISCA 1997.
- José Luis Abellán, Juan Fernández, Manuel E. Acacio, GLocks: Efficient Support for Highly-Contended Locks in Many-Core CMPs, IPDPS2011.

(25 points)

4.2 List-based Queue Locks

Start with the program from exercise 3.3. Now, implement the lock primitive using the MCS lock algorithm from Mellor-Crummey/Scott 1991 (Algorithms for Scalable Synchronization on Shared-Memory Multiprocessors).

Develop your programs and perform initial testing on one of the **creek** nodes. Validate the correctness of these two new programs with the same methodology as in exercise 2.3. I.e., for a varying **C** and **N**, ensure that after execution the counter matches **C**, i.e. there are no race conditions anymore.

(35 points)

4.3 Lock performance analysis

Now, measure the overall execution time using suitable functions (for instance *clock_gettime* or *gettimeofday*). Report the overall execution time and the derived number of updates per second for a varying number of threads (1-48) and sufficiently large number of updates (providing stable results). For this experiment, use the computer **moore** (48 cores in total). As **moore** is often heavily used, please ensure that you only use it for performance experiments.

	1. PTHREAD MUTEX		2. ATOMIC INCREMENT		3. LOCK_RMW		4. QUEUE LOCK	
Thread count	**Execution time**	**Updates per second**	**Execution time**	**Updates per second**	**Execution time**	**Updates per second**	**Execution time**	**Updates per second**
1								
2								
4								
8								
12								
16								
24								
32								
40								
48								

Re-use the data for experiments 1.-3. Run the experiment for the MCS Lock and report results. Include a graphical representation here (varying number of threads on x-axis with updates per second on y-axis):

(15 points)

Total: 75 points