An Experimental Comparison of Concurrent Data Structures

Mark Gibson

**Introduction: (Thin)**

**My Work:**

**Context:**

**Structure:**

**Results:**

**Literature Review: (Fatish)**

**What Motivated You?**

**Set the Scene**

**Produce a Critique**

**Method: (Fat)**

**What do you have to do?**

**How will you do it?**

**Experiments & Evaluation: (Fatish)**

**Does your method work?**

**Afterword: (Thin)**

**What happened?**

**Lessons learnt?**

Read The Art of Multiprocessor Programming by Maurice Herlihy & Nir Shavit

Read several articles on Locklessinc.com/articles

Including Spinlocks and Read Write locks

23/01/14

Agreed with David Gregg to implement a Ring Buffer as the first data structure. I decided to go for a simple design in C to begin with. I implemented it using pthreads and two modes of operation; the first used mutexes in a simple lock and unlock fashion while the second used a spinlock. However, the implementation of the spinlock proved tricky due to the low level nature of C and so I decided early on to start using C++, in order to access the higher level intrinsics, since my focus was not on the implementation but rather the testing of these structures.

<http://people.csail.mit.edu/edya/publications/OptimisticFIFOQueue-DISC2004.pdf> --Gives details on implementing a buffer using CAS.

27/01/14

Researched several papers on the topic of concurrent data structures, placed them into the research folder

Amended the output of the program to be easier to paste into excel for graph generation

Gathered data from stoker for both locked and spinlock modes of ring buffer

Need to gather data from my machine, spoon and ducss