

Praktikum angewandte Systemsoftwaretechnik: Vergleich nichtblockierender Queues

Michael Banken Lorenz Haspel

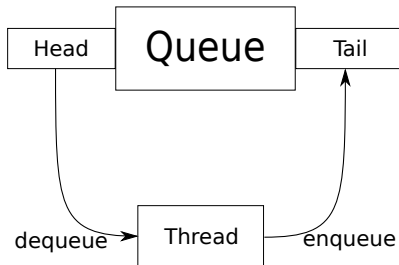
Friedrich-Alexander Universität Erlangen-Nürnberg

20.9.2013



- im Rahmen von LAOS/OctoPOS
- Queues als zentrale Datenstruktur für die Verwaltung von Threads
- paralleler Zugriff auf gemeinsame Daten
- unsere Zielsetzung: Performanzanalyse und die Entwicklung paralleler Tests mit verschiedenen nichtblockierenden Queues



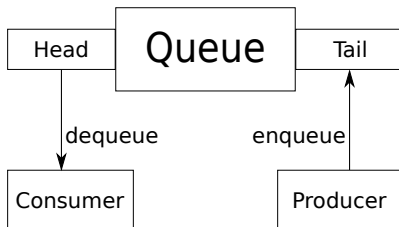


- Eine Queue, zu Beginn gefüllt
- Jeder Thread nimmt ein Element aus der Queue und fügt es wieder ein
- Elemente bleiben konstant, werden also immer wieder verwendet



- Versuchsreihen mit variierender Anzahl an Threads und Elementen
- feste Gesamtzahl an Operationen
- Zeit wird gemessen
- drei Testreihen:
 - halb so viele Elemente wie Threads
 - genau so viele Elemente wie Threads
 - doppelt so viele Elemente wie Threads





- Multiple Producers Multiple Consumers
- Queue zu Beginn leer
- Producerthreads fügen Elemente ein
- Consumerthreads nehmen Elemente raus
- jedes Element wird nur einmal verwendet



- Versuchsreihe mit variierender Anzahl an Threads
- feste Anzahl an Elementen, die die Queue durchlaufen
- zwei Testreihen:
 - variable Zahl an Producerthreads, ein Consumerthread
 - gleiche Zahl von Producern wie Consumern

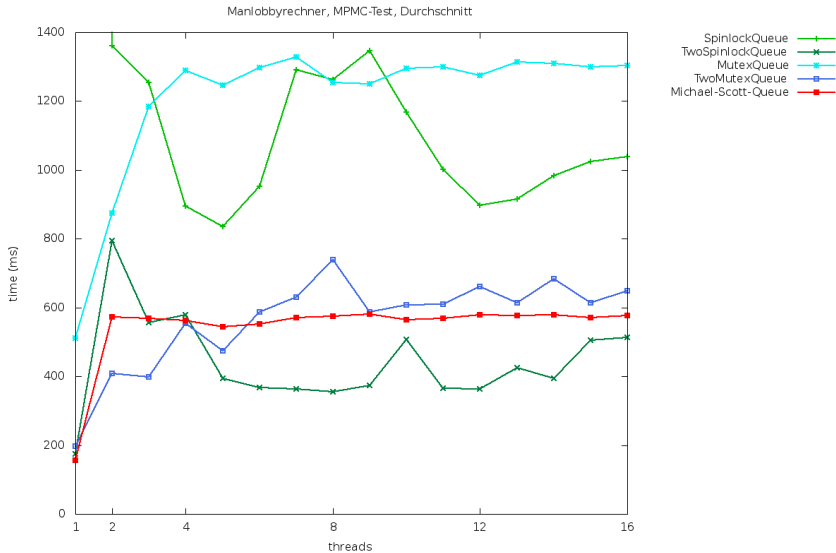


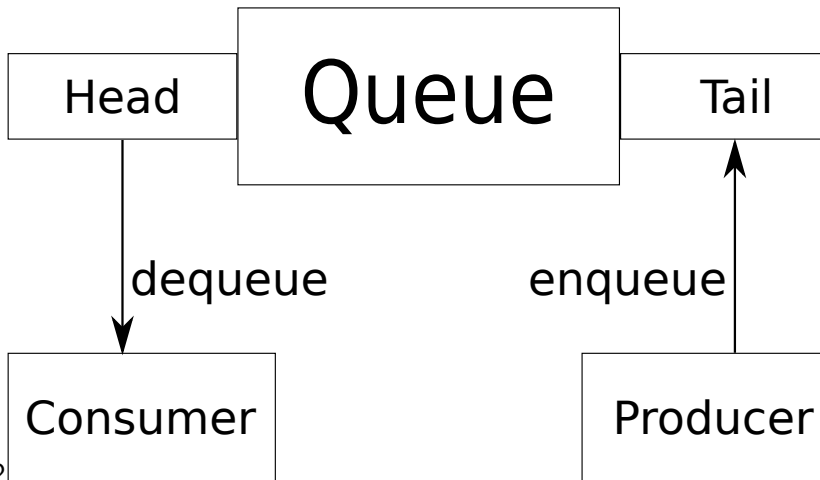
- Core i7 aus der Manlobby:
 - 4 Kerne, 8 durch Hyperthreading
 - Tests neben normaler Betriebslast (Programme andere User, X-server, Ubuntu-Hintergrundprogramme)
- Haswellrechner "fastbox":
 - 8 Kerne, Intel-Haswell Prozessoren
 - Test von Queues mit naiver Implementierung von Haswell-Transaktionen
 - Locks auf Prozessorebene
- 48-Kern Rechner „bigbox“:
 - Tests bei massiv parallelem Zugriff



- SimpleQueue: Queue mit einem globalen Lock für en- und dequeue-Operationen
 - Varianten: Spinlock, Mutex, Haswell-Transaktion
- TwoLockQueue: Queue mit zwei Locks, je eines für en- und dequeue Operationen
 - Varianten: Spinlock, Mutex, Haswell-Transaktionen
- Michael-Scott-Queue: nichtblockierende Queue
- WnbsQueue: Queue aus den BST-Folien vom Lehrstuhl
 - keine Tests durchgeführt, Probleme der verwendeten Implementierung bei parallelem Zugriff
- MPSC-Queue: Multiple-Producers-Single-Consumer-Queue
 - spezialisierte nicht blockierende Queue







/2/2



- empty



Vielen Dank für Ihre Aufmerksamkeit!

