ari ‘aith’ iramanesh

Synchronous and Asynchronous Concurrent Producer-Consumer Queues

	syncQueue	asyncQueue	queue8	queue8API
Average Turnaround Time (sec)	2.524	2.506	2.517	1.251

Because the tests are single producer and single consumer, I expected to see little difference in the turnaround times between the different implementations. It turned out that the times were almost identical, except for queue8API. Given how different the implementations are, line-for-line, it was a surprising result for me.

syncQueue was designed for a single producer and consumer, since a single dequeue operation turns off the flag for further consecutive dequeues, regardless of how many enqueues have been performed. The asyncQueue is also designed for a single producer and consumer, which wastes its purpose, in my opinion. It uses a circular buffer to store and load information, but the spinloop conditions for list full-ness and emptiness are not robust for 2+ consumers & producers:

```
void enq()...
// list is full
while ((head.load() + 1) % CQUEUE_SIZE == tail.load() % Q_SIZE);
int reserved_index = atomic_fetch_add(&head, 1) % Q_SIZE;
```

Multiple producers could pass this spinloop at the same time, and both increment the head past the tail, despite being full. My proposed solution is to add a flag to ensure the passage of one producer at a time:

```
void enq()...
```

```
bool acquired = false;
while (!acquired) {
	acquired = true;
	while ((head.load() + 1) % CQUEUE_SIZE == tail.load() %
CQUEUE_SIZE);
	acquired = atomic_compare_exchange_weak(&flag, acquired, false);
}
int reserved_index = atomic_fetch_add(&head,1) % CQUEUE_SIZE;
...
flag.store(true);

private:
	atomic<bool> flag;
```

However, this increases the complexity of code substantially. Since it was not needed to successfully execute the code, I left it out... but it’s something to consider!

queue8 received no noticeable speedup. This was surprising to me, especially given the speedup of queue8API. I attribute the lack of speedup to queue8 still being subject to the same amount of branch prediction overhead as syncQueue and asyncQueue. queue8API turns 8 if-statements into 1 for both enqueue and dequeue!