**Main Contributions:**

The paper introduced a memory type called transactional memory, this memory can make programmers define customized read-modify write operations that apply to multiple, independently chosen words of memory. Author did a full simulation on this memory, and get a good result on it by basically out match all the lock-based memory technique, it’s very impressive. Using this kind of memory will simplify the implementations of data structures inside algorithms with lock-free memory, which free of multiple problems the lock-based memory had, and achieve great performance.

**My Thoughts:**

The first thing I noticed is they actually put some limitations at the rational part. The paper talked about the memory’s 2 disadvantages, one is the set size problem without mechanisms to handle overflow, another one is the cache size should be large, but they used a small one. It’s only the 4th part, right before the result part. This makes the paper extremely convincing.

Second thing is their result. The result part shows this memory is fantastic compare to all other lock-based memories, it is way more effective on basically every task. They did mention some limitation like frequent barriers would make it slower, but still it’s so good compare to other memories.

The last thing is a bit personal, but I think the structure of this paper lack a bit order, the second part is the structure introduction part, but there are a few usages, and very little big map about the memory. The forth part is rationale, just before the result comes up. I think this is alright, but it’s a bit unclear if read though the whole paper.

**Positive Point:**

The performance of this thing is impressive

**Negative Point:**

The structure is a bit off meta.