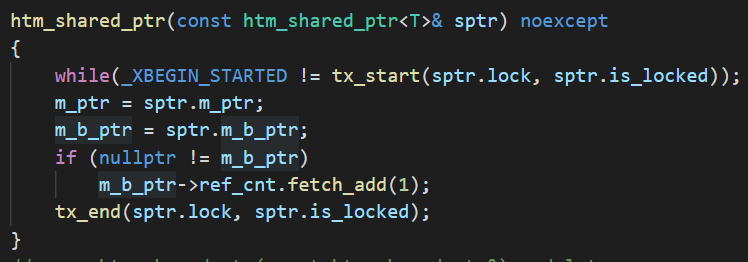
**HTM atomic shared pointer Fallback Path**

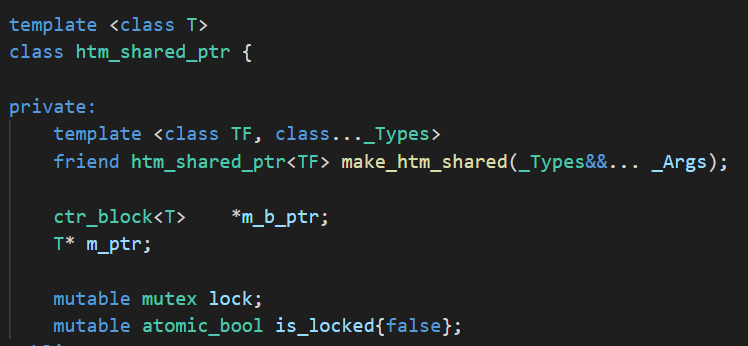
**1. 구현 설명**

Fallback path에서 lock을 걸고 진행하는데도 data race가 발생했다. 원인을 찾아보니 lock을 걸 때 자기 자신의 m\_ptr과 m\_b\_ptr만을 보호하기 때문에 복사 생성자나 대입연산자에서 target의 m\_ptr과 m\_b\_ptr을 읽을 때 중간 값을 읽는 경우가 발생했다.

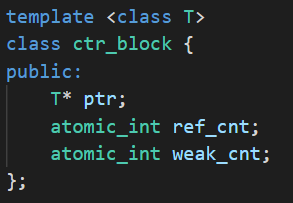
따라서 target을 &로 받는 복사 생성자에서는 fallback path로 넘어갈 때 또는 htm transaction을 진행할 때 target 내부의 lock과 is\_locked에 접근했다.



이때 복사 생성자에서는 target을 const로 받아야 하는데 target의 is\_locked와 mutex에 접근해야 하기 때문에 두 member data를 mutable keyword로 선언했다.

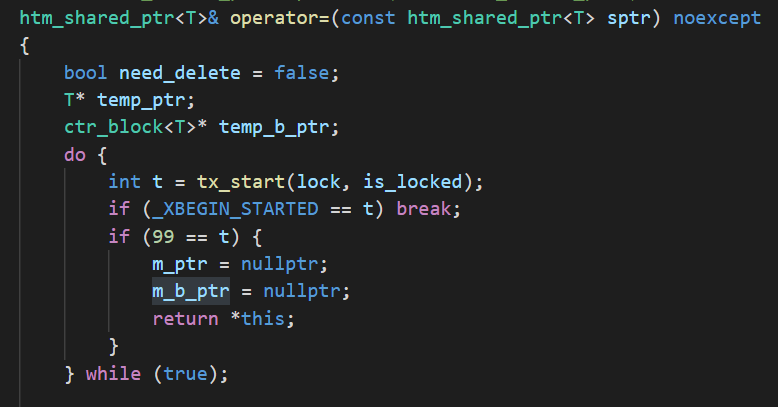


또한 fallback path로 접근할 때 control block에 여러 곳에서 동시에 접근가능하기 때문에 ref\_cnt를 atomic\_int로 선언했다.



복사 생성자에서는 this의 m\_ptr과 m\_b\_ptr은 lock으로 보호할 필요가 없기 때문에 this의 lock은 걸지 않았다.

그러나 대입 연산자에서는 target을 &로 받을 때 target의 lock과 this의 lock 둘 다 걸어야 하기 때문에 target을 &로 받지 않아 복사 생성되도록 했다. 이렇게 중간에 복사 생성되면 복사생성 된 htm\_shared\_ptr가 copy하려는 data를 가리키고 있어서 ref\_cnt를 1이하로 못 내려가게 만들어 target의 m\_ptr과 m\_b\_ptr이 가리키고 있는 data가 안전하다. 또한 복사 생성된 htm\_shared\_ptr은 호출하는 thread만이 접근가능 하기 때문에 data race로 인한 중간 값 걱정 없이 읽을 수 있다. 그러나 소멸자 overhead가 따로 발생하기 때문에 target을 &로 받아 lock을 두 번 거는 방법으로 최적화했다. 하지만 가독성만 나빠지고 별로 성능차이가 없었다. (제출하는 코드는 최적화하지 않은 코드)



**2. 성능 측정**

**- 40000번 테스트 소요 시간 (ms)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | lazy | Atomic load/store | Mutex | HTM | HTM+ fallback  (부정확) | HTM2 | **HTM2 + fallback** |
| 1 thread | 22 | 411 | 676 | 673 | 654 | 2896 | **3280** |
| 2 thread | 19 | 504 | 3020 | 1327 | 342 | 3381 | **1654** |
| 4 thread | 16 | 402 | 1413 | 675 | 185 | 1333 | **862** |
| 8 thread | 15 | 366 | 5153 | 6313 | 147 | 1796 | **859** |
| 16 thread | 15 | 370 | 4196 | 4631 | 152 | 1905 | **884** |

**- abort 횟수**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | 1 thread | 2 thread | 4 thread | 8 thread | 16 thread |
| total aborts | 6086 | 387633 | 1278896 | 19620420 | 19636165 |
| capacity aborts | 8 | 196 | 595 | 9817 | 9520 |
| conflict aborts | 686 | 379979 | 1269408 | 19599752 | 19617329 |
| forced aborts | 0 | 0 | 0 | 0 | 0 |
| explicit aborts | 0 | 0 | 0 | 0 | 0 |
| rest aborts | 5392 | 7458 | 8893 | 10851 | 9316 |
| total commits | 68854438 | 67924751 | 67139452 | 68544537 | 68543121 |
| **total fallback** | **0** | **4789** | **20934** | **103494** | **100836** |