|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | 싱글 스레드 | Coarse Stack | LockFreeStack | LFBOSTACK |
| 1Thread | 1139ms | 1207 | 813 | 790 |
| 2Thread |  | 1358 | 633 | 723 |
| 4Thread |  | 1677 | 920 | 791 |
| 8Thread |  | 2143 | 1181 | 880 |
| 16Thread |  | 2138 | 1191 | 882 |

CoarseStack의 성능이 안좋은원인

1. 코어끼리 캐시 버스 오버헤드가 생겨서
2. mutex오버헤드

* Lock Free 스택은 병렬성이 없다 .왜냐하면 모든 쓰레드가 Top에서 push,pop하기 때문에
* BackOff

1. CAS 를 실패했을 경우 잠시 대기해서 버스 부하에 영향을 주지 않게 하는 것.
2. 미리 자신이 CAS를 실패할것을 쓰레드는 알 수 없다.
3. 쓰레드 개수가 많을수록 실패했을 때 다시 실패할 확률이 더 커짐
4. maxDelay = 최대 기다리는 시간을 정해서 Starvation을 막아야함

\*\_asm RDTSC; -> ReadSystemClock 🡺 시스템안에 시간을 읽어라

int delay;

if (limit != 0)

delay = rand()%limit+1;

else limit = 1;

if(limit<maxDelay)limit = 2\*limit;

int start, current;

\_asm RDTSC;

\_asm mov start, eax;

do {

\_asm RDTSC;

\_asm mov current, eax;

} while (current – start < delay)

* 운영체제 호출이 없어서 Sleep\_for해서 운영체제 호출한것보다 더 성능이 좋음
* 그러나 루프에서 메모리를 계속 접근
* LFBACKOFF 알고리즘은 LOCKFREE에 성능을 높이는것이아니라 성능이 떨어지는 것을 보완해주는 알고리즘이다
* minDelay 는 latency까지 고려해야함
* 큐보다 스택에서 ABA문제가 더 잘 생긴다.
* 64BIT CAS는 ATOMIC\_COMAPARE\_EXCHANGE\_STRONG 은 제공하지 않음
* WINAPI나 ASM명령어를 사용해야함
* 그러나 오버헤드가 많아서 잘 사용하지 않음
* Free-List

Free\_LIST()

{

first = new Node(0);

second = new Node(0);

}

* First에 넣고 second에서 빼고
* exchange를 해서 용도를 서로 바꿈
* Stack 이나 Queue는 프로그램의 메인이 아니기 때문에 성능이 크게 중요하지 않다
* 만약 stack,queue를 메인으로 써야해서 성능을 올려야 한다면