

Author: Arthur Sonzogni

View: public

Comment: all@chromium.org all@google.com everyone@google.com (+ask)

This is meant to answer question:

https://chromium-review.googlesource.com/c/chromium/src/+/3463442/3..5/base/allocator/partition_allocator/partition_ref_count.h#b109

Bartek:

Well, not really. When `ENABLE_DANGLING_RAW_PTR_CHECKS` is on `uint64_t` will be used, even for 32-bit architectures. This means we need the special, 64-bit wide CAS instructions that Benoit mentioned. Hopefully the compiler is smart enough to generate those for us, but please check.

Reproducer:

Since `PartitionRefCount::Acquire()` is always inlined, I had to add:

```
__attribute__((noinline))  
in order to find it.
```

```
__attribute__((noinline)) void Acquire() {  
    CheckCookie();  
    auto old_count = count_.fetch_add(kRawPtrInc, std::memory_order_relaxed);  
    // Check overflow.  
    PA_CHECK((old_count & kRawPtrCountMask) != kRawPtrCountMask);  
}
```

Then, the assembly is extracted using:

```
objdump -d ./out/Debug/base_unittests --visualize-jumps=extended-color  
| grep "PartitionRefCount.*Acquire" -A100 --color=always  
| tee output
```

Reproducer:

x86_64

Args

Result:

x86

Args

Result:

arm

Extra steps

Args

Result:

arm64

Extra steps

Args

Result:

x86_64

Args

```
use_goma = true
enable_nacl = false
is_debug = false
dcheck_always_on = true
use_backup_ref_ptr = true
use_partition_alloc = true
enable_dangling_raw_ptr_checks = true
put_ref_count_in_previous_slot = true
```

Result:

000000001184c70 <PartitionRefCount::Acquire()>:		
1184c70:	55	push %rbp
1184c71:	48 89 e5	mov %rsp,%rbp
1184c74:	b8 02 00 00 00	mov \$0x2,%eax
1184c79:	f0 48 0f c1 07	lock xadd %rax, (%rdi)
1184c7e:	f7 d0	not %eax
1184c80:	a9 fe ff ff ff	test \$0xfffffffffe,%eax
1184c85:	/-- 74 02	je 1184c89 <PartitionRefCount::Acquire()+0x19>
1184c87:	5d	pop %rbp
1184c88:	c3	ret
1184c89:	\-> 48 8d 3d dc ed 03 ff	lea -0xfc1224(%rip),%rdi # 1c3a6c <.L.str.1348>
1184c90:	5d	pop %rbp
1184c91:	e9 1a 8c 1d 00	jmp 135d8b0 <_ZN7logging8RawCheckEPKc>
1184c96:	cc	int3
1184c97:	cc	int3
1184c98:	cc	int3
1184c99:	cc	int3
1184c9a:	cc	int3
1184c9b:	cc	int3
1184c9c:	cc	int3
1184c9d:	cc	int3
1184c9e:	cc	int3
1184c9f:	cc	int3

x86

Args

```
use_goma = true
enable_nacl = false
is_debug = false
dcheck_always_on = true
use_backup_ref_ptr = true
use_partition_alloc = true
enable_dangling_raw_ptr_checks = true
put_ref_count_in_previous_slot = true
target_cpu = "x86"
```

Result:

```
01093b70 <PartitionRefCount::Acquire()>:
1093b70:      55                push    %ebp
1093b71:      89 e5             mov     %esp,%ebp
1093b73:      53                push    %ebx
1093b74:      57                push    %edi
1093b75:      56                push    %esi
1093b76:      83 ec 0c          sub     $0xc,%esp
1093b79:  /-- e8 00 00 00 00 call    1093b7e <PartitionRefCount::Acquire()+0xe>
1093b7e:  \-> 5e                pop     %esi
1093b7f:      81 c6 ee be 64 00 add     $0x64beee,%esi
1093b85:      8b 7d 08          mov     0x8(%ebp),%edi
1093b88:      8b 07             mov     (%edi),%eax
1093b8a:      8b 57 04          mov     0x4(%edi),%edx
1093b8d:      0f 1f 00          nopl    (%eax)
1093b90:  /-> 89 c3             mov     %eax,%ebx
1093b92:  | 83 c3 02          add     $0x2,%ebx
1093b95:  | 89 d1             mov     %edx,%ecx
1093b97:  | 83 d1 00          adc     $0x0,%ecx
1093b9a:  | f0 0f c7 0f       lock cmpxchg8b (%edi)
1093b9e:  \-- 75 f0           jne     1093b90 <PartitionRefCount::Acquire()+0x20>
1093ba0:      f7 d0             not     %eax
1093ba2:      a9 fe ff ff ff    test    $0xffffffff,%eax
1093ba7:  /-- 74 08           je      1093bb1 <PartitionRefCount::Acquire()+0x41>
1093ba9:  /--|-> 83 c4 0c          add     $0xc,%esp
1093bac:  | | 5e                pop     %esi
1093bad:  | | 5f                pop     %edi
1093bae:  | | 5b                pop     %ebx
1093baf:  | | 5d                pop     %ebp
1093bb0:  | | c3                ret
1093bb1:  | \-> 8d 86 24 5a 9d fe lea     -0x162a5dc(%esi),%eax
1093bb7:  | 89 04 24          mov     %eax,(%esp)
1093bba:  | 89 f3             mov     %esi,%ebx
1093bbc:  | e8 cf 5b 1b 00    call    1249790 <_ZN7logging8RawCheckEPKc>
1093bc1:  \----- eb e6           jmp     1093ba9 <PartitionRefCount::Acquire()+0x39>
1093bc3:      cc                int3
1093bc4:      cc                int3
1093bc5:      cc                int3
1093bc6:      cc                int3
```

arm

Extra steps

```
sudo apt install binutils-arm-linux-gnueabi
```

And then:

```
arm-linux-gnueabi-hf-objdump -d ./out/Debug/lib.unstripped/libbase_unittests__library.so
--visualize-jumps=extended-color
| grep "PartitionRefCount.*Acquire" -A100 --color=always
| tee output
```

Args

```
use_goma = true
enable_nacl = false
is_debug = false
dcheck_always_on = true
use_backup_ref_ptr = true
use_partition_alloc = true
enable_dangling_raw_ptr_checks = true
put_ref_count_in_previous_slot = true
target_cpu="arm"
target_os="android"
```

Result:

```
00832064 <PartitionRefCount::Acquire()>:
 832064:      b510      push {r4, lr}
 832066: /-> e8d0 127f  ldrexld    r1, r2, [r0]
 83206a: |   1c8b      adds r3, r1, #2
 83206c: |   f142 0200  adc.w      r2, r2, #0
 832070: |   e8c0 3274  strexd     r4, r3, r2, [r0]
 832074: |   2c00      cmp  r4, #0
 832076: \-- d1f6      bne.n      832066 <PartitionRefCount::Acquire()+0x2>
 832078:      43c8      mvns r0, r1
 83207a:      0840      lsrs r0, r0, #1
 83207c:      bf18      it   ne
 83207e:      bd10      popne     {r4, pc}
 832080:      4802      ldr  r0, [pc, #8] ; (83208c <PartitionRefCount::Acquire()+0x28>)
 832082:      4478      add  r0, pc
 832084:      e8bd 4010  ldmia.w    sp!, {r4, lr}
 832088:      f0c2 bcb6  b.w  8f49f8 <_ZN7logging8RawCheckEPKc>
 83208c:      ff8d28dd  .word     0xff8d28dd
```

arm64

Extra steps

I ended up using **llvm-objdump** instead.

```
./third_party/llvm-build/Release+Asserts/bin/llvm-objdump -d
./out/Debug/lib.unstripped/libbase_unittests__library.so
| grep "PartitionRefCount.*Acquire" -A200 --color=always
| tee output
```

Args

```
use_goma = true
enable_nacl = false
is_debug = false
dcheck_always_on = true
use_backup_ref_ptr = true
use_partition_alloc = true
enable_dangling_raw_ptr_checks = true
put_ref_count_in_previous_slot = true
target_cpu="arm64"
target_os="android"
```

Result:

```
000000000d719f0 <PartitionRefCount::Acquire()>:
d719f0: 08 7c 5f c8 /----> ldxr x8, [x0]
d719f4: 09 09 00 91 |      add x9, x8, #2
d719f8: 09 7c 0a c8 |      stxr w10, x9, [x0]
d719fc: aa ff ff 35 \----- cbnz w10, 0xd719f0 <PartitionRefCount::Acquire()>
d71a00: e8 03 28 2a      mvn w8, w8
d71a04: 1f 79 7f f2      tst x8, #0xfffffffffe
d71a08: 40 00 00 54 /----- b.eq 0xd71a10 <PartitionRefCount::Acquire()+0x20>
d71a0c: c0 03 5f d6 |      ret
d71a10: 80 a3 ff 90 \-----> adrp x0, 0x1e1000 <base::internal::PartitionAllocTest_Bookkeeping_Test::TestBody()+0x38c0>
d71a14: 00 24 18 91      add x0, x0, #1545
d71a18: a8 c9 04 14      b 0xea40b8 <_ZN7logging8RawCheckEPKc>
```