C/C++中的 `#define` 指令用于创建宏。宏本质上是一种文本替换机制，由预处理器在编译前执行。当您 `#define` 某物时，预处理器将替换定义的宏的每个出现，用其对应值或代码。

在提供的代码中， `FF` 、 `GG` 、 `HH` 和 `II` 被定义为宏，而不是函数。这就是为什么它们没有显式的 `return` 语句。它们不是返回值，而是直接使用 `+=` 和 `=` 操作符修改它们的参数（ `a` 、 `b` 、 `c` 、 `d` ）。这些更改直接应用于传递给宏的变量。这与函数有重要区别，因为函数通常在参数的副本上操作（除非你传递指针或引用）。

FF 宏定义了 MD5 算法中第一轮变换的步骤，其中：

- `a`, `b`, `c`, `d` 是 MD5 状态的四个 32 位字
- `x` 是输入消息的一个 32 位块
- `s` 指定左循环移位的位数
- `ac` 是一个基于正弦函数的常量

guess > test.o

```
bba46eb8b53cf65d50ca54b2f8afd9db
原始MD5Hash结果: bba46eb8b53cf65d50ca54b2f8afd9db
验证结果: 相同

Authorized users only. All activities may be monitored and reported.

```

得到了正确的结果

效率变低了，我们需要分析原因：

内存分配和访问模式低效：

- 为每个消息单独调用StringProcess，导致多次内存分配
- 从多个不连续内存位置加载数据，导致缓存利用率低

```
Guesses generated: 9853408
Guesses generated: 10106852
Guess time:8.23305seconds
Hash time:13.8043seconds
Train time:98.4926seconds

Authorized users only. All activities may be monitored and reported.

```

训练时间太久了所以我们将训练数据减少到30000

```
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:18.7103seconds
Hash time:9.05261seconds
Train time:0.856026seconds

Authorized users only. All activities may be monitored and reported.

```

```
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:18.6308seconds
Hash time:8.94697seconds
Train time:0.857836seconds

Authorized users only. All activities may be monitored and reported.

```

重新选定基准值9.05s，8.94s，下面是并行化之后的结果（已经优化了访存模式

```
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:19.6023seconds
Hash time:13.2519seconds
Train time:0.846007seconds

Authorized users only. All activities may be monitored and reported.

```

```
Annotate MD5Hash_SIMD
Zoom into main thread
Zoom into main DSO (use the 'k' hotkey to zoom directly into the kernel)
Browse map details
Run scripts for samples of symbol [MD5Hash_SIMD]
Run scripts for all samples
Switch to another data file in PWD
Exit
```

1. **Annotate MD5Hash_SIMD**
   - 显示MD5Hash_SIMD函数的源代码级性能注解
   - 可以看到每行代码执行的频率和耗时，帮助定位具体的性能瓶颈
2. **Zoom into main thread**
   - 只查看主线程的性能数据
   - 过滤掉其他线程的数据
3. **Zoom into main DSO**
   - 缩小视图到主要的动态共享对象
   - `k`快捷键可以直接查看内核相关性能数据
4. **Browse map details**
   - 浏览内存映射的详细信息
   - 查看代码和数据的内存分布
5. **Run scripts for samples of symbol [MD5Hash_SIMD]**
   - 对MD5Hash_SIMD函数的采样数据运行自定义脚本
   - 可以进行进一步的自动化分析
6. **Run scripts for all samples**
   - 对所有采样数据运行脚本
7. **Switch to another data file in PWD**

- 切换到当前目录中的其他perf数据文件

8. **Exit**
   - 退出perf report工具

```
Samples: 191K of event 'cycles:u', 4000 Hz, Event count (approx.): 71032229743, DSO: main
MD5Hash_SIMD  /home/s2313211/guess/main [Percent: local period]
Percent │        lsl   x19, x0, #2
        │        mov   x0, x19
        │      → bl    _init
        │        mov   x2, x0
        │        mov   x1, x2
        │        sub   x0, x19, #0x1
   4.56 │ 17c:   cmp   x0, #0x0
        │      ↓ b.lt  194
   0.05 │        strb  wzr, [x1]
        │        add   x1, x1, #0x1
   0.02 │        sub   x0, x0, #0x1
        │      ↑ b     17c
```

汇编代码中与0相比较耗费了我们较多的时间，所以我们选择打开循环，直接执行循环里面的操作

```
for (int i = 0; i < 4; i++) {
    Byte* block_ptr = paddedMessages[i] + block * 64 + j * 4;
    // 使用位运算而不是多次左移减少指令
    values[i] = (uint32_t)block_ptr[0] |
        ((uint32_t)block_ptr[1] << 8) |
        ((uint32_t)block_ptr[2] << 16) |
        ((uint32_t)block_ptr[3] << 24);
}
```

```
// 展开的数据加载循环
Byte* block_ptr0 = paddedMessages[0] + block * 64 + j * 4;
Byte* block_ptr1 = paddedMessages[1] + block * 64 + j * 4;
Byte* block_ptr2 = paddedMessages[2] + block * 64 + j * 4;
Byte* block_ptr3 = paddedMessages[3] + block * 64 + j * 4;

values[0] = (uint32_t)block_ptr0[0] | ((uint32_t)block_ptr0[1] << 8) |
((uint32_t)block_ptr0[2] << 16) | ((uint32_t)block_ptr0[3] << 24);
values[1] = (uint32_t)block_ptr1[0] | ((uint32_t)block_ptr1[1] << 8) |
((uint32_t)block_ptr1[2] << 16) | ((uint32_t)block_ptr1[3] << 24);
values[2] = (uint32_t)block_ptr2[0] | ((uint32_t)block_ptr2[1] << 8) |
((uint32_t)block_ptr2[2] << 16) | ((uint32_t)block_ptr2[3] << 24);
values[3] = (uint32_t)block_ptr3[0] | ((uint32_t)block_ptr3[1] << 8) |
((uint32_t)block_ptr3[2] << 16) | ((uint32_t)block_ptr3[3] << 24);
```

```
paddedMessages[0] = buffer + 0 * max_padded_length;
PrepareMessage(inputs[0], paddedMessages[0], &messageLengths[0]);
paddedMessages[1] = buffer + 1 * max_padded_length;
PrepareMessage(inputs[1], paddedMessages[1], &messageLengths[1]);
paddedMessages[2] = buffer + 2 * max_padded_length;
PrepareMessage(inputs[2], paddedMessages[2], &messageLengths[2]);
paddedMessages[3] = buffer + 3 * max_padded_length;
PrepareMessage(inputs[3], paddedMessages[3], &messageLengths[3]);
```

```
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:19.6311seconds
Hash time:12.5921seconds
Train time:0.845521seconds

Authorized users only. All activities may be monitored and reported.

```

```
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:19.6759seconds
Hash time:12.6927seconds
Train time:0.855539seconds

Authorized users only. All activities may be monitored and reported.

```

```
Guesses generated: 10097691
Guess time:19.7199seconds
Hash time:12.5602seconds
Train time:0.853786seconds

Authorized users only. All activities may be monitored and reported.

```

差不多优化了1秒，还是很多的

我们使用静态变量`a0_init`，减少每次进入函数时的vdupq_n_u32运算

```
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:19.7208seconds
Hash time:12.8036seconds
Train time:0.854116seconds

Authorized users only. All activities may be monitored and reported.

```

好像没什么变化

```
// 替换这样的循环:
max_length = std::max(max_length, inputs[0].length());
max_length = std::max(max_length, inputs[1].length());
max_length = std::max(max_length, inputs[2].length());
max_length = std::max(max_length, inputs[3].length());

// 使用条件移动代替条件分支
max_padding_bits = (max_padded_bits == 448) ? 512 :
                   ((448 - max_padded_bits) & 0x1FF);
```

```
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:19.7885seconds
Hash time:12.6222seconds
Train time:0.848716seconds

Authorized users only. All activities may be monitored and reported.
```

还是没有什么优化感觉

```
const size_t block_offsets[16] = {0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40, 44,
48, 52, 56, 60};
```

再加一个静态变量，把这个的乘法也省掉了

```
Byte* block_ptr0 = paddedMessages[0] + block * 64 + block_offsets[j];
Byte* block_ptr1 = paddedMessages[1] + block * 64 + block_offsets[j];
Byte* block_ptr2 = paddedMessages[2] + block * 64 + block_offsets[j];
Byte* block_ptr3 = paddedMessages[3] + block * 64 + block_offsets[j];
```

```
Guesses generated: 9796757
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:19.6745seconds
Hash time:12.7869seconds
Train time:0.842985seconds

Authorized users only. All activities may be monitored and reported.

```

还是没变化，我们从头再来哈，这个是并行的结果19.46s

```
问题    输出    调试控制台    终端    端口                                perf - guess
Samples: 167K of event 'cycles:u', Event count (approx.): 87800961180
Overhead  Command  Shared Object          Symbol
  19.28%  main     main                   [.] MD5Hash_SIMD
   8.39%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   7.75%  main     main                   [.] ROTATELEFT_SIMD
   4.64%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_copy
   3.16%  main     main                   [.] PT::operator=
   3.01%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_da
   2.08%  main     main                   [.] F_SIMD
   2.04%  main     main                   [.] std::vector<int, std::allocator<int> >::_M_move_assign
   1.96%  main     main                   [.] G_SIMD
   1.86%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vector
   1.77%  main     main                   [.] H_SIMD
   1.60%  main     main                   [.] std::vector<int, std::allocator<int> >::operator=
Tip: Add -I to perf record to sample register values, which will be visible in perf report sample context.
```

13.78s

```
Samples: 145K of event 'cycles:u', Event count (approx.): 76020724029
Overhead  Command  Shared Object          Symbol
  23.38%  main     main                   [.] MD5Hash
   9.44%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   5.24%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_cop
   3.74%  main     main                   [.] PT::operator=
   3.40%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_d
   2.33%  main     main                   [.] std::vector<int, std::allocator<int> >::_M_move_assign
   2.08%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vecto
   1.80%  main     main                   [.] std::allocator<int>::allocator
   1.79%  main     main                   [.] std::vector<int, std::allocator<int> >::operator=
   1.61%  main     main                   [.] StringProcess
   1.44%  main     main                   [.] std::vector<int, std::allocator<int> >::~vector
   1.43%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::get_allocator
Tip: See assembly instructions with percentage: perf annotate <symbol>
```

```
Samples: 143K of event 'cycles:u', Event count (approx.): 76376600377
Overhead  Command  Shared Object          Symbol
  23.35%  main     main                   [.] MD5Hash
   9.54%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   5.24%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_cop
   3.69%  main     main                   [.] PT::operator=
   3.41%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_d
   2.32%  main     main                   [.] std::vector<int, std::allocator<int> >::_M_move_assign
   2.11%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vecto
   1.75%  main     main                   [.] std::allocator<int>::allocator
   1.75%  main     main                   [.] std::vector<int, std::allocator<int> >::operator=
   1.51%  main     main                   [.] StringProcess
   1.42%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::get_allocator
   1.39%  main     main                   [.] std::vector<int, std::allocator<int> >::~vector
   1.38%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::~_Vector_base
   1.26%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_base
   1.10%  main     main                   [.] std::__do_alloc_on_move<std::allocator<int> >
   1.08%  main     libstdc++.so.6.0.28    [.] std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >
   1.07%  main     main                   [.] std::allocator<int>::~allocator
   1.04%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl
   0.98%  main     main                   [.] std::vector<segment, std::allocator<segment> >::operator=
   0.97%  main     main                   [.] std::vector<segment, std::allocator<segment> >::_M_move_assign
   0.96%  main     main                   [.] std::vector<int, std::allocator<int> >::vector
   0.93%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_deallocate
   0.92%  main     main                   [.] std::__alloc_on_move<std::allocator<int> >
   0.89%  main     libc.so.6              [.] 0x000000000009c080
   0.86%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_swap_data
   0.82%  main     main                   [.] __gnu_cxx::new_allocator<int>::~new_allocator
   0.81%  main     main                   [.] std::vector<segment, std::allocator<segment> >::~vector
   0.80%  main     main                   [.] std::allocator<segment>::allocator
   0.80%  main     main                   [.] std::_Destroy<int*>
   0.77%  main     main                   [.] std::move<std::allocator<int>&>
   0.77%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator
   0.77%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::~_Vector_base
   0.71%  main     main                   [.] std::_Destroy_aux<false>::__destroy<segment*>
   0.67%  main     main                   [.] std::__do_alloc_on_move<std::allocator<segment> >
   0.64%  main     main                   [.] std::allocator<segment>::~allocator
   0.63%  main     main                   [.] std::_Destroy<segment*>
   0.62%  main     main                   [.] std::__alloc_on_move<std::allocator<segment> >
   0.61%  main     main                   [.] std::__copy_move_backward<true, false, std::random_access_iterator_tag>::__copy
   0.58%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::get_allocator
   0.54%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_swa
   0.53%  main     main                   [.] std::__copy_move<true, false, std::random_access_iterator_tag>::__copy_m<PT*, P
   0.53%  main     main                   [.] std::move<std::vector<int, std::allocator<int> >&>
   0.52%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl::_Vector_imp
   0.52%  main     main                   [.] std::vector<segment, std::allocator<segment> >::vector
   0.52%  main     libc.so.6              [.] malloc
   0.49%  main     main                   [.] __gnu_cxx::new_allocator<segment>::~new_allocator
```

来个长图比对一下

```
Samples: 165K of event 'cycles:u', Event count (approx.): 88185461034
Overhead  Command  Shared Object        Symbol
  19.18%  main     main                 [.] MD5Hash_SIMD
   8.45%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   7.78%  main     main                 [.] ROTATELEFT_SIMD
   4.81%  main     main                 [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_copy
   3.17%  main     main                 [.] PT::operator=
   2.90%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_da
   2.09%  main     main                 [.] std::vector<int, std::allocator<int> >::_M_move_assign
   2.02%  main     main                 [.] F_SIMD
   1.98%  main     main                 [.] G_SIMD
   1.88%  main     main                 [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vector
   1.74%  main     main                 [.] H_SIMD
   1.58%  main     main                 [.] std::vector<int, std::allocator<int> >::operator=
   1.51%  main     main                 [.] I_SIMD
   1.50%  main     main                 [.] std::allocator<int>::allocator
   1.34%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::~_Vector_base
   1.27%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::get_allocator
   1.22%  main     main                 [.] std::vector<int, std::allocator<int> >::~vector
   1.22%  main     main                 [.] PrepareMessage
   1.00%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_Vector_base
   0.98%  main     main                 [.] std::__do_alloc_on_move<std::allocator<int> >
   0.96%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl
   0.90%  main     main                 [.] std::allocator<int>::~allocator
   0.88%  main     main                 [.] std::vector<segment, std::allocator<segment> >::operator=
   0.86%  main     main                 [.] std::_Destroy<int*>
   0.83%  main     main                 [.] std::vector<int, std::allocator<int> >::vector
   0.83%  main     main                 [.] std::vector<segment, std::allocator<segment> >::_M_move_assign
   0.83%  main     main                 [.] std::_Vector_base<segment, std::allocator<segment> >::~_Vector_base
   0.82%  main     main                 [.] __gnu_cxx::new_allocator<int>::~new_allocator
   0.82%  main     main                 [.] std::__alloc_on_move<std::allocator<int> >
   0.78%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_M_deallocate
   0.76%  main     libstdc++.so.6.0.28  [.] std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >:
   0.76%  main     main                 [.] std::allocator<segment>::allocator
   0.75%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_swap_data
   0.73%  main     main                 [.] std::_Destroy_aux<false>::__destroy<segment*>
   0.72%  main     main                 [.] std::vector<segment, std::allocator<segment> >::~vector
   0.70%  main     main                 [.] std::move<std::allocator<int>&>
   0.68%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator
   0.65%  main     main                 [.] std::_Destroy<segment*>
   0.65%  main     main                 [.] std::__do_alloc_on_move<std::allocator<segment> >
   0.61%  main     main                 [.] std::allocator<segment>::~allocator
   0.59%  main     main                 [.] std::__alloc_on_move<std::allocator<segment> >
   0.52%  main     main                 [.] std::move<std::vector<int, std::allocator<int> >&>
   0.51%  main     main                 [.] std::_Vector_base<segment, std::allocator<segment> >::get_allocator
   0.51%  main     main                 [.] std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator
   0.50%  main     main                 [.] std::__copy_move<true, false, std::random_access_iterator_tag>::__copy_m<PT*, PT
   0.48%  main     main                 [.] std::__copy_move_backward<true, false, std::random_access_iterator_tag>::__copy_
Tip: To add Node.js USDT(User-Level Statically Defined Tracing): perf buildid-cache --add `which node`
```

豪德我们选中了内联函数ROTATELEFT_SIMD,并将其变为宏定义，时间直接减少了1s

```
// ROTATELEFT的SIMD版本
inline bit32x4_t ROTATELEFT_SIMD(bit32x4_t x, int n) {
    return vorrq_u32(vshlq_n_u32(x, n), vshrq_n_u32(x, 32 - n));
}

#define ROTATELEFT_SIMD(num, n) (vorrq_u32(vshlq_n_u32((num), (n)),
vshrq_n_u32((num), 32 - (n))))
```

```
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:30.8781seconds
Hash time:18.0347seconds
Train time:1.43003seconds
[ perf record: Woken up 58 times to write data ]
[ perf record: Captured and wrote 14.952 MB perf.data (322126 samples) ]
[s2313211@master_ubss1 guess]$ perf report
```

```
Samples: 161K of event 'cycles:u', Event count (approx.): 86843961874
Overhead  Command  Shared Object          Symbol
  24.95%  main     main                   [.] MD5Hash_SIMD
   8.32%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   4.82%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_copy
   3.22%  main     main                   [.] PT::operator=
   2.99%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_da
   2.38%  main     main                   [.] F_SIMD
   2.18%  main     main                   [.] std::vector<int, std::allocator<int> >::_M_move_assign
   1.99%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vector
   1.92%  main     main                   [.] G_SIMD
   1.76%  main     main                   [.] H_SIMD
   1.73%  main     main                   [.] I_SIMD
   1.62%  main     main                   [.] std::allocator<int>::allocator
   1.61%  main     main                   [.] std::vector<int, std::allocator<int> >::operator=
   1.29%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::~_Vector_base
   1.28%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::get_allocator
   1.22%  main     main                   [.] PrepareMessage
   1.21%  main     main                   [.] std::vector<int, std::allocator<int> >::~vector
   1.00%  main     main                   [.] std::__do_alloc_on_move<std::allocator<int> >
   0.99%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_base
   0.96%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl
   0.95%  main     libstdc++.so.6.0.28    [.] std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >:
   0.94%  main     main                   [.] std::allocator<int>::~allocator
   0.89%  main     main                   [.] std::__alloc_on_move<std::allocator<int> >
   0.89%  main     main                   [.] std::vector<int, std::allocator<int> >::vector
   0.86%  main     main                   [.] std::vector<segment, std::allocator<segment> >::operator=
   0.85%  main     main                   [.] __gnu_cxx::new_allocator<int>::~new_allocator
   0.85%  main     main                   [.] std::_Destroy<int*>
   0.85%  main     main                   [.] std::vector<segment, std::allocator<segment> >::_M_move_assign
   0.79%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_deallocate
   0.79%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::~_Vector_base
   0.79%  main     libc.so.6              [.] 0x000000000009c080
   0.76%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_swap_data
   0.75%  main     main                   [.] std::vector<segment, std::allocator<segment> >::~vector
   0.74%  main     main                   [.] std::move<std::allocator<int>&>
```

ROTATELEFT也从高时间消失了

那我们不妨把这些也改为宏定义

```
// 添加SIMD版本的基本函数
inline bit32x4_t F_SIMD(bit32x4_t x, bit32x4_t y, bit32x4_t z) {
  return vorrq_u32(vandq_u32(x, y), vandq_u32(vmvnq_u32(x), z));
}

inline bit32x4_t G_SIMD(bit32x4_t x, bit32x4_t y, bit32x4_t z) {
  return vorrq_u32(vandq_u32(x, z), vandq_u32(y, vmvnq_u32(z)));
}

inline bit32x4_t H_SIMD(bit32x4_t x, bit32x4_t y, bit32x4_t z) {
  return veorq_u32(veorq_u32(x, y), z);
}

inline bit32x4_t I_SIMD(bit32x4_t x, bit32x4_t y, bit32x4_t z) {
  return veorq_u32(y, vorrq_u32(x, vmvnq_u32(z)));
}

#define F_SIMD(x, y, z) (vorrq_u32(vandq_u32((x), (y)), vandq_u32(vmvnq_u32((x)),
(z))))
#define G_SIMD(x, y, z) (vorrq_u32(vandq_u32((x), (z)), vandq_u32((y),
vmvnq_u32((z)))))
#define H_SIMD(x, y, z) (veorq_u32(veorq_u32((x), (y)), (z)))
#define I_SIMD(x, y, z) (veorq_u32((y), vorrq_u32((x), vmvnq_u32((z)))))
```

```
Guesses generated: 9595702
Guesses generated: 9696752
Guesses generated: 9796757
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:30.9011seconds
Hash time:17.4475seconds
Train time:1.42324seconds
[ perf record: Woken up 56 times to write data ]
[ perf record: Captured and wrote 14.753 MB perf.data (318200 samples) ]
```

时间也来到了17.4s

```
问题   输出   调试控制台   终端   端口                                   perf - guess
Samples: 159K of event 'cycles:u', Event count (approx.): 85256999212
Overhead  Command  Shared Object          Symbol
  32.57%  main     main                   [.] MD5Hash_SIMD
   8.63%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   4.90%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_copy
   3.25%  main     main                   [.] PT::operator=
   3.05%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_da
   2.20%  main     main                   [.] std::vector<int, std::allocator<int> >::_M_move_assign
   2.00%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vector
   1.64%  main     main                   [.] std::allocator<int>::allocator
   1.64%  main     main                   [.] std::vector<int, std::allocator<int> >::operator=
   1.31%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::~_Vector_base
   1.28%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::get_allocator
   1.24%  main     main                   [.] std::vector<int, std::allocator<int> >::~vector
   1.19%  main     main                   [.] PrepareMessage
   1.05%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_base
   1.02%  main     main                   [.] std::__do_alloc_on_move<std::allocator<int> >
   1.00%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl
   0.89%  main     main                   [.] std::_Destroy<int*>
   0.89%  main     main                   [.] std::vector<segment, std::allocator<segment> >::operator=
   0.88%  main     main                   [.] std::vector<int, std::allocator<int> >::vector
   0.88%  main     main                   [.] std::allocator<int>::~allocator
   0.88%  main     main                   [.] std::__alloc_on_move<std::allocator<int> >
   0.87%  main     main                   [.] std::vector<segment, std::allocator<segment> >::_M_move_assign
   0.85%  main     libstdc++.so.6.0.28    [.] std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >:
   0.83%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::~_Vector_base
   0.82%  main     main                   [.] __gnu_cxx::new_allocator<int>::~new_allocator
   0.81%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_deallocate
   0.75%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_swap_data
   0.75%  main     main                   [.] std::move<std::allocator<int>&>
   0.74%  main     main                   [.] std::allocator<segment>::allocator
   0.71%  main     main                   [.] std::_Destroy<segment*>
   0.71%  main     main                   [.] std::vector<segment, std::allocator<segment> >::~vector
   0.70%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator
   0.69%  main     main                   [.] std::_Destroy_aux<false>::__destroy<segment*>
   0.65%  main     main                   [.] std::allocator<segment>::~allocator
   0.62%  main     main                   [.] std::__do_alloc_on_move<std::allocator<segment> >
   0.59%  main     main                   [.] std::__alloc_on_move<std::allocator<segment> >
   0.53%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::get_allocator
   0.53%  main     main                   [.] std::move<std::vector<int, std::allocator<int> >&>
   0.50%  main     main                   [.] std::vector<segment, std::allocator<segment> >::vector
   0.50%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_get_Tp_allocator
Tip: To count events in every 1000 msec: perf stat -I 1000
```

内联函数的时间消耗已经被消除了但是主函数的时间增加了，我们可以看出内联函数虽然可以减少主函数的时间，但自身会增加运行时间

我们观察就可以发现凹其他的时间已经没用了，只能去凹主函数

```
问题   输出   调试控制台   终端   端口                                   perf - guess
Samples: 159K of event 'cycles:u', 3250 Hz, Event count (approx.): 84785084261, Thread: main
MD5Hash_SIMD  /home/s2313211/guess/main [Percent: local period]
Percent          mov   x0, x19
               → bl    _init
                 mov   x2, x0
                 mov   x1, x2
                 sub   x0, x19, #0x1
   4.05  1d8:    cmp   x0, #0x0
   0.00        ↓ b.lt  1f0
   0.06          strb  wzr, [x1]
                 add   x1, x1, #0x1
   0.01          sub   x0, x0, #0x1
               ↑ b     1d8
   0.02  1f0:    str   x2, [sp, #20072]
Press 'h' for help on key bindings
```

看到这个没有，百分之4.05，优化他！

```
memset(paddedMessage + length + 1, 0, paddingBytes - 1); // 填充0字节是这个语句
```

```
static Byte zero_buffer[MAX_BUFFER_SIZE] = {0};//加上一个静态缓冲区
memcpy(output + input_length + 1, zero_buffer, padding_bytes - 1);//用拷贝复制信息
```

```
Guesses generated: 9490827
Guesses generated: 9595702
Guesses generated: 9696752
Guesses generated: 9796757
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:28.6479seconds
Hash time:16.6006seconds
Train time:1.26677seconds
[ perf record: Woken up 46 times to write data ]
[ perf record: Captured and wrote 11.763 MB perf.data (252457 samples) ]
```

```
Guesses generated: 9595702
Guesses generated: 9696752
Guesses generated: 9796757
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:28.5686seconds
Hash time:16.555seconds
Train time:1.29609seconds
[ perf record: Woken up 45 times to write data ]
[ perf record: Captured and wrote 11.741 MB perf.data (252128 samples) ]
```

16.6s了已经

```
Percent │        and   x0, x0, #0xffffffffffffffc0
  0.11  │        str   x0, [sp, #20080]
        │        ldr   x0, [sp, #20080]
        │        lsl   x19, x0, #2
        │        mov   x0, x19
        │      → bl    _init
        │        mov   x2, x0
        │        mov   x1, x2
        │        sub   x0, x19, #0x1
  4.00  │ 1d8:   cmp   x0, #0x0
        │      ↓ b.lt  1f0
  0.05  │        strb  wzr, [x1]
        │        add   x1, x1, #0x1
  0.01  │        sub   x0, x0, #0x1
        │      ↑ b     1d8
  0.03  │ 1f0:   str   x2, [sp, #20072]
        │        sub   x0, sp, #0x120
        │        add   x0, x0, #0x2c0
  0.12  │        ldr   x1, [sp, #20072]
        │        str   x1, [x0]
        │        sub   x0, sp, #0x120
        │        add   x0, x0, #0x2c0
  0.02  │        ldr   x0, [x0]
        │        add   x1, sp, #0x190
        │        mov   x2, x1
        │        mov   x1, x0
```

性能瓶颈的汇编代码，我们再定位一下代码位置

```
Byte* buffer = new Byte[4 * max_padded_length]();  // 括号导致内存清零

Byte* buffer = new Byte[4 * max_padded_length];//只初始化必要的部分
```

```
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:28.7946seconds
Hash time:15.6209seconds
Train time:1.31909seconds
[ perf record: Woken up 45 times to write data ]
[ perf record: Captured and wrote 11.554 MB perf.data (249221 samples) ]
[s2313211@master ubss1 guess]$
```

15.5s

```
问题    输出    调试控制台    终端    端口                                                    perf - guess
Samples: 124K of event 'cycles:u', 2750 Hz, Event count (approx.): 84619676314
0.02          ldrb  w0, [x0]
              lsl   w0, w0, #16
              orr   w1, w1, w0
0.12          ldr   x0, [sp, #20024]
              add   x0, x0, #0x3
0.04          ldrb  w0, [x0]
0.01          lsl   w0, w0, #24
              orr   w0, w1, w0
0.17          str   w0, [sp, #308]
0.01          add   x0, sp, #0x128
0.03          str   x0, [sp, #19920]
0.16          ldr   x0, [sp, #19920]
0.17          ldr   q0, [x0]
0.03          nop
1.31          add   x0, sp, #0x4, lsl #12
              ldrsw x0, [x0, #3708]
              lsl   x0, x0, #4
              add   x1, sp, #0x20
0.00          str   q0, [x1, x0]
              add   x0, sp, #0x4, lsl #12
0.01          ldr   w0, [x0, #3708]
              add   w0, w0, #0x1
              add   x1, sp, #0x4, lsl #12
0.10          str   w0, [x1, #3708]
            ↑ b     394
0.00    5d4:  ldr   q0, [sp, #20160]
0.01          str   q0, [sp, #20000]
              ldr   q0, [sp, #20144]
              str   q0, [sp, #19984]
              ldr   q0, [sp, #20128]
0.01          str   q0, [sp, #19968]
              ldr   q0, [sp, #20112]
              str   q0, [sp, #19952]
              ldr   q0, [sp, #19984]
0.00          str   q0, [sp, #19696]
              ldr   q0, [sp, #19968]
```

定位代码

```
// 更高效的数据加载
for (int j = 0; j < 16; j++) {
    // 一次性加载4个值
    uint32_t values[4];
    // 展开的数据加载循环
    Byte* block_ptr0 = paddedMessages[0] + block * 64 + j * 4;
    Byte* block_ptr1 = paddedMessages[1] + block * 64 + j * 4;
    Byte* block_ptr2 = paddedMessages[2] + block * 64 + j * 4;
    Byte* block_ptr3 = paddedMessages[3] + block * 64 + j * 4;

    values[0] = (uint32_t)block_ptr0[0] | ((uint32_t)block_ptr0[1] << 8) |
((uint32_t)block_ptr0[2] << 16) | ((uint32_t)block_ptr0[3] << 24);
    values[1] = (uint32_t)block_ptr1[0] | ((uint32_t)block_ptr1[1] << 8) |
((uint32_t)block_ptr1[2] << 16) | ((uint32_t)block_ptr1[3] << 24);
    values[2] = (uint32_t)block_ptr2[0] | ((uint32_t)block_ptr2[1] << 8) |
((uint32_t)block_ptr2[2] << 16) | ((uint32_t)block_ptr2[3] << 24);
    values[3] = (uint32_t)block_ptr3[0] | ((uint32_t)block_ptr3[1] << 8) |
((uint32_t)block_ptr3[2] << 16) | ((uint32_t)block_ptr3[3] << 24);
    // 使用NEON指令直接加载数据
    M[j] = vld1q_u32(values);
}
```

```
static const size_t BLOCK_OFFSETS[16] = {0, 4, 8, 12, 16, 20, 24, 28, 32, 36, 40,
44, 48, 52, 56, 60};
//加上静态变量避免重复计算
```

```
Guesses generated: 9796757
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:28.9344seconds
Hash time:15.3386seconds
Train time:1.27717seconds
[ perf record: Woken up 43 times to write data ]
[ perf record: Captured and wrote 11.504 MB perf.data (247907 samples) ]
[s2313211@master_ubss1 guess]$
```

似乎没有优化，现在最多的变为了这个

```
  0.03          ldr   w0, [sp, #300]
  0.02          rev   w0, w0
  0.14          str   w0, [sp, #300]
  0.01          ldr   w0, [sp, #304]
  0.00          rev   w0, w0
  0.09          str   w0, [sp, #304]
  0.02          ldr   w0, [sp, #308]
  0.01          rev   w0, w0
  0.14          str   w0, [sp, #308]
                add   x0, sp, #0x128
  0.02          str   x0, [sp, #19920]
  0.15          ldr   x0, [sp, #19920]
  0.14          ldr   q0, [x0]
  0.22          nop
  1.63          add   x0, sp, #0x4, lsl #12
                ldrsw x0, [x0, #3692]
                lsl   x0, x0, #4
                add   x1, sp, #0x20
  0.02          str   q0, [x1, x0]
                add   x0, sp, #0x4, lsl #12
  0.06          ldr   w0, [x0, #3692]
  0.02          add   w0, w0, #0x1
                add   x1, sp, #0x4, lsl #12
  0.05          str   w0, [x1, #3692]
              ↑ b     3a0
        4b0:    ldr   q0, [sp, #20144]
                str   q0, [sp, #20000]
                ldr   q0, [sp, #20128]
  0.00          str   q0, [sp, #19984]
```

问题　输出　调试控制台　终端　端口　　perf - guess

```
Samples: 122K of event 'cycles:u', Event count (approx.): 84194169277
Overhead  Command  Shared Object          Symbol
  31.00%  main     main                   [.] MD5Hash_SIMD
   8.87%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_copy_data
   5.42%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_M_copy
   3.29%  main     main                   [.] PT::operator=
   2.97%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_Vector_impl_da
   1.94%  main     main                   [.] std::vector<int, std::allocator<int> >::_M_move_assign
   1.86%  main     main                   [.] std::allocator<int>::allocator
   1.65%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::_Vector_impl_data::_Vector
   1.52%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::get_allocator
   1.47%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::~_Vector_base
   1.46%  main     main                   [.] std::vector<int, std::allocator<int> >::operator=
   1.24%  main     main                   [.] std::vector<int, std::allocator<int> >::~vector
   1.13%  main     main                   [.] PrepareMessage
   1.12%  main     main                   [.] std::__do_alloc_on_move<std::allocator<int> >
   1.11%  main     main                   [.] std::_Destroy<int*>
   1.08%  main     main                   [.] std::vector<segment, std::allocator<segment> >::_M_move_assign
   1.02%  main     main                   [.] std::allocator<int>::~allocator
   1.00%  main     main                   [.] std::__alloc_on_move<std::allocator<int> >
   0.89%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_base
   0.89%  main     main                   [.] __gnu_cxx::new_allocator<int>::~new_allocator
   0.89%  main     main                   [.] std::allocator<segment>::allocator
   0.89%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl_data::_M_swap_data
   0.88%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_Vector_impl::_Vector_impl
   0.88%  main     main                   [.] std::vector<segment, std::allocator<segment> >::~vector
   0.85%  main     main                   [.] std::vector<segment, std::allocator<segment> >::operator=
   0.82%  main     main                   [.] std::move<std::allocator<int>&>
   0.79%  main     main                   [.] std::_Vector_base<int, std::allocator<int> >::_M_deallocate
   0.77%  main     main                   [.] std::_Destroy_aux<false>::__destroy<segment*>
   0.74%  main     main                   [.] std::_Vector_base<segment, std::allocator<segment> >::get_allocator
   0.73%  main     libstdc++.so.6.0.28    [.] std::__cxx11::basic_string<char, std::char_traits<char>, std::allocator<char> >:
   0.68%  main     main                   [.] std::_Destroy<segment*>
```

```
Guesses generated: 9897171
Guesses generated: 9997458
Guesses generated: 10097691
Guess time:20.0164seconds
Hash time:10.9214seconds
Train time:0.8663seconds

Authorized users only. All activities may be monitored and reported.
```