

## 圖(一)

```
Samples: 16K of event 'cycles', 4000 Hz, Event count (approx.): 2780101174
Overhead
          Shared Object
                                                  Symbol
                                                     _Z3AddPiS_S_
_Z4copyPiS_
           fib
          fib
                                                  [k] native_queued_spin_lock_slowpath
          [kernel]
                                                  [k] page_vma_mapped_walk
          [kernel]
   0.47%
          [kernel]
                                                     native_write_msr
   0.34%
          [kernel]
                                                  [k] copy_page_regs
                                                      __strcmp_sse2_unaligned
   0.25%
          libc-2.27.so
   0.25%
          [kernel]
                                                      _raw_spin_lock
                                                     page_referenced_one
   0.24%
          [kernel]
                                                  [.] g_hash_table_lookup
   0.22%
          libglib-2.0.so.0.5600.3
          libglib-2.0.so.0.5600.3
                                                   ] g_slice_alloc
   0.21%
   0.21%
          libc-2.27.so
                                                     malloc
   0.20%
          [kernel]
                                                     vma_interval_tree_subtree_search
   0.18%
          [kernel]
                                                     vmw_cmdbuf_header_submit
                                                     update blocked averages
   0.18%
          [kernel]
   0.17%
                                                     vma_interval_tree_iter_next
          [kernel]
   0.16%
          [kernel]
                                                     native_apic_msr_eoi_write
   0.15%
                                                      get_page_from_freelist
          [kernel]
```

圖(二)

由圖(一)可見 main 函式消耗的效能最多,而其中的 fib()又占了 99.90%, Fib()主要由 Add()和 copy()所構成,兩者的 overhead 幾乎各半,此外,由圖二可見,

第一列:符號引發的性能事件的比例,默認指佔用的 CPU 週期比例,

第二列:符號所在的 DSO (動態共享對象),可以是應用程序,內核,動態鏈接庫,模塊,

前兩項的紅色部分皆為 fib(),其所耗的 cpu 高達 67%(overhead),由此可知 fib() 為造成效能瓶頸的主要元兇。