

# CPU性能優化的幾個思路

陳龍星&曾義格



# 回顧CPU性能指標

- 用戶CPU使用率(user/nice)
- 系統CPU使用率(不含中斷)
- 等待I/O的CPU使用率iowait
- Interrupt  
『軟/硬 中斷呼叫』  
的CPU使用率
- Load average  
(每N秒的CPU使用率)
  - Process count
- Proccess Context switch  
(上下文切換)
  - 自願/非自願
- CPU cache hit rate (hit ratio)



# 三個問題之一：

- 如何判斷現在要做的優化是否有效？優化後，能提升多少性能？
  - 確認性能的量化指標
  - 優化前量測
  - 優化後量測
    - 包含副作用的測量



# 三個問題之二：

多個性能問題同時發生時，要先優化哪一個？

- 最先達上限的資源指標
- 找出影響最大的點
- 找出因果關係



# 三個問題之三：

- 當有多種優化方法可以選擇時，要選擇哪一種？
  - 優化不代表沒有成本
    - Data Plane Development Kit ( <https://www.dpdk.org> )  
10G net / 2010 open source/ intel Nehalem 2c2t ~ 2c4t ~ 6c6t(Xeon) ~4c8t amd k7 4c4t (max 2 cpu)  
( <https://medium.com/@atoonk/building-a-high-performance-linux-based-traffic-generator-with-dpdk-93bb9904416c> )
  - 切勿因小失大



# 三個問題之三：

- 當有多種優化方法可以選擇時，要選擇哪一種？
  - 別笑想一步登天
  - 並不是所有問題都值得優化
  - 先做已知沒副作用的
  - 做最簡單最快速並且其他指標副作用不會達上限的。



# 量化指標

- 應用/服務
  - 吞吐量
  - 請求延遲（回應速度）
    - Ex: java GC , G1GC , ZGC
- 系統資源
  - CPU使用率



# 量化指標

- 系統資源
  - RAM使用率
  - I/O使用率
  - swap使用量/使用率
- 正確性
- 安全性
- 穩定性





# 漸進式優化

- ~~系統~~/系統環境
  - 命令/腳本
  - 環境參數
- 定時腳本
- 應用/~~執行~~參數
- 編譯參數
- ~~kernel~~優化
- 程式碼
  -
- 服務架構



# 漸進式優化

- ~~系統~~/系統環境

- 命令/腳本

- ~~cpu~~綁定
    - ~~cpu~~獨佔
    - ~~NUMA~~
    - ~~renice~~

- cgroups

- Year 2010 : 200  
line code

- swappiness

- Server
  - android
  - Max?min?zram?



# 漸進式優化

- ~~系統~~/系統環境

- 環境參數

- Irqbalance
    - smp\_affinity
    - JAVA\_OPTS
      - -server
      - -XX:  
+IgnoreUnrecognizedVMOptions
      - -XX:  
+UnlockExperimentalVMOptions
      - -Xms2m
      - -Xmx2g

- --add-modules=ALL
  - -SYSTEM
  - -XX:+AggressiveOpts
  - -XX:+UseG1GC
  - -  
Dosgi.dataAreaRequiresExplicitInit=  
true
  - -XX:PermSize=2m
  - -XX:MetaspaceSize=2m
  - -Xss256k
  - -XX:MaxHeapFreeRatio=10
  - -XX:MinHeapFreeRatio=5



- -XX:-ShrinkHeapInSteps
- -XX:+ScavengeBeforeFullGC
- -XX:+UseBiasedLocking
- -XX:BiasedLockingStartupDelay=0
- -XX:+EliminateLocks
- -XX:UseAVX=2
- -XX:UseSSE=4
- -XX:+UseSSE42Intrinsics
- -XX:+UseTLAB
- -XX:+ResizeTLAB
- -XX:+UseAdaptiveGCBoundary
- -XX:+UseAdaptiveSizePolicy
- -XX:+BackgroundCompilation
- -XX:CompileThreshold=8000
- -Djava.net.preferIPv4Stack=true
- -XX:InitiatingHeapOccupancyPercent=0
- -XX:G1ReservePercent=10
- -Dosgi.requiredJavaVersion=1.6
- -XX:GCTimeRatio=1
- -Duser.language=zh
- -Duser.country=TW
- -XX:ReservedCodeCacheSize=256m
- -XX:+TieredCompilation
- -XX:+UseFastAccessorMethods
- -XX:+UseFastEmptyMethods
- -XX:+UseFastJNIAccessors
- -XX:+UseCompressedOops
- -XX:+BindGCThreadThreadsToCPUs
- -XX:+DoEscapeAnalysis
- -XX:MaxGCPauseMillis=200
- -XX:+UseStringDeduplication



# 漸進式優化

- 定時腳本
  - Swapfile縮放
- 應用/執行參數
  - nice
- 編譯參數
  - march
  - mtune
- kernel優化
  - 非官方sources/patch
    - CK(MuQSS) / RT / openvZ / xen / tuxonice / hardened / zen
    - PF
  -



# 漸進式優化

- 程式碼
  - for / try / if / while
  - C# : Parallel.For / PLINQ
  - C# / java : volatile
  - Buffered I/O
  - 線程
  -
- async
- SIMD
- 算法
  - $O(n^2)$  的排序算法（如冒泡、插入排序等） $\Rightarrow$   $O(n \log n)$  的排序算法（如快排、歸併排序等）
- 架構
  - actor
- 服務架構



# 漸進式優化

- 系統/系統環境

- 命令/腳本

- ~~cpu~~綁定
    - ~~cpu~~獨佔
    - NUMA
    - ~~renice~~
    - cgroups
      - Year 2010 : 200 line code
    - swappiness
      - Server
      - android
      - Max?min?zram?

- 環境參數

- irqbalance
    - smp\_affinity
    - JAVA\_OPTS

- 定時腳本

- Swapfile縮放

- 應用/執行參數

- nice

- 編譯參數

- march
    - mtune

- kernel優化

- 非官方sources/patch
      - CK(MuQSS) / RT / openvZ / xen / tuxonice / hardened / zen
      - PF

- 程式碼

- for / try / if / while
    - C# : Parallel.For / PLINQ
    - C# / java : volatile
    - Buffered I/O
    - 線程
    - async
    - SIMD
    - 算法
      - $O(n^2)$  的排序算法（如冒泡、插入排序等） $\Rightarrow O(n\log n)$  的排序算法（如快排、歸併排序等）
    - 架構
      - actor

- 服務架構



# 漸進式優化

- 系統/系統環境

- 命令/腳本

- ~~cpu~~綁定
    - ~~cpu~~獨佔
    - NUMA
    - ~~renice~~
    - cgroups
      - Year 2010 : 200 line code
    - swappiness
      - Server
      - android
      - Max?min?zram?

- 環境參數

- irqbalance
    - smp\_affinity
    - JAVA\_OPTS

- 定時腳本

- Swapfile縮放

- 應用/執行參數

- nice

- 編譯參數

- march
  - mtune

- kernel優化

- 非官方sources/patch
    - CK(MuQSS) / RT / openvZ / xen / tuxonice / hardened / zen
    - PF

- 程式碼

- for / try / if / while
  - C# : Parallel.For / PLINQ
  - C# / java : volatile
  - Buffered I/O
  - 線程
  - async
  - SIMD
  - 算法
    - $O(n^2)$  的排序算法（如冒泡、插入排序等） $\Rightarrow O(n \log n)$  的排序算法（如快排、歸併排序等）
  - 架構
    - actor

- 服務架構





# GCC編譯參數輔助功能

- `gcc -march=native -Q --help=target`

The following options are target specific:

<code>-m128bit-long-double</code>	<code>[disabled]</code>
<code>-m16</code>	<code>[disabled]</code>
<code>-m32</code>	<code>[enabled]</code>
<code>-m3dnow</code>	<code>[disabled]</code>
<code>-m3dnowa</code>	<code>[disabled]</code>
<code>-m64</code>	<code>[disabled]</code>
<code>-m80387</code>	<code>[enabled]</code>
<code>-m8bit-idiv</code>	<code>[disabled]</code>
<code>-m96bit-long-double</code>	<code>[enabled]</code>
<code>-mabi=</code>	<code>sysv</code>
<code>-mabm</code>	<code>[enabled]</code>
<code>-maccumulate-outgoing-args</code>	<code>[disabled]</code>
<code>-maddress-mode=</code>	<code>short</code>
<code>-madx</code>	<code>[enabled]</code>
<code>-maes</code>	<code>[enabled]</code>
<code>-malign-data=</code>	<code>compat</code>
<code>-malign-double</code>	<code>[disabled]</code>
<code>-malign-functions=</code>	<code>0</code>
<code>-malign-jumps=</code>	<code>0</code>
<code>-malign-loops=</code>	<code>0</code>
<code>-malign-stringops</code>	<code>[enabled]</code>
<code>-mandroid</code>	<code>[disabled]</code>



# GCC輔助功能

- `gcc -march=native -Q --help=target`

Known code models (for use with the `-mmodel=` option):

`32 kernel large medium small`

Valid arguments to `-mfpmath=`:

`387 387+sse 387,sse both sse sse+387 sse,387`

Known data alignment choices (for use with the `-malign-data=` option):

`abi cacheline compat`

Known vectorization library ABIs (for use with the `-mveclibabi=` option):

`acml svml`

Known address mode (for use with the `-maddress-mode=` option):

`long short`

Known stack protector guard (for use with the `-mstack-protector-guard=` option):

`global tls`

Valid arguments to `-mstringop-strategy=`:

`byte_loop libcall loop rep_4byte rep_8byte rep_byte unrolled_loop vector_loop`

# GCC Optimize-Options

- <https://gcc.gnu.org/onlinedocs/gcc-10.1.0/gcc/Optimize-Options.html#Optimize-Options>



# • GCC Instrumentation-Options 檢測選項

- Meltdown & Spectre 幽靈漏洞
  - <https://zh.wikipedia.org/wiki/%E5%B9%BD%E7%81%B5%E6%BC%8F%E6%B4%9E>
- <https://gcc.gnu.org/onlinedocs/gcc-10.1.0/gcc/Optimize-Options.html>
  - -mmitigate-rop
  - ~~-Wchkp~~
  - -fchkp-treat-zero-dynamic-size-as-infinite
  - -fstack-protector-all
  - -fstack-check



# julia程式碼SIMD優化

<https://medium.com/@dboyliao/parallel-computing-with-julia-part-i-optimization-cbf0d88e2c34>

```
function simd_sum(data)
    result = zero(eltype(data[1]))
    @simd for idx in eachindex(data)
        @inbounds result += data[idx]
    end
    result
end
```



# Java

- SIMD指令需要循環100000次
- 
- Java Jit永遠不會自動對浮點進行矢量化處理  
這是因為重新排列浮點運算會改變結果。  
 $0.1 + (0.2 + 0.3) \neq (0.1 + 0.2) + 0.3$



# Actor

- akka / scala / Erlang / Elixir

