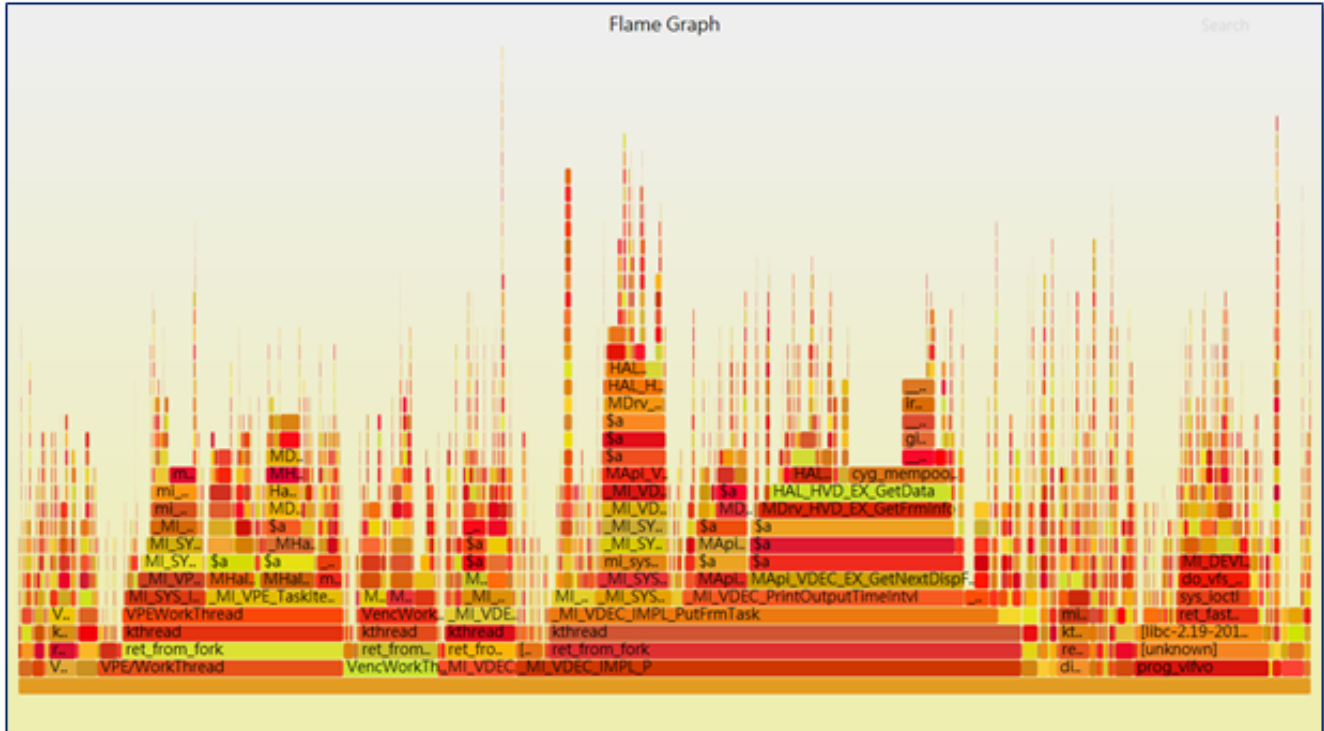


系統性能分析利器-Pref & Flame Graph 介紹和使用

Flame Graph可以提供視覺化的性能分析能力，以SVG格式呈現，可縮放查看局部，方便快速定位出CPU效能瓶頸。Flame Graph統計了採樣周期內，code-path被執行的時間比例，縱軸是code-path，從下至上一般就是call stack。橫軸是某個stack的占用時間，跨度越大說明其占CPU比重越高，也就是最耗費CPU的，stack在橫軸上是按照字母序排列的，顏色深淺僅僅是為了區分，並沒特殊意義。



以下將說明Flame Graph產生流程：

1. 用perf擷取系統執行狀態

- Step 1.1: 在EVB上執行perf record

```
# ./perf record -a -g sleep 10  
  
-a 表示cpu  
-o 指定輸出文件，若未指定則自動在當前目錄下存檔名為perf.data  
-g 紀錄call stack  
sleep 指定持續時間，單位為秒
```

若執行時顯示

表示系統缺少GCC runtime library,

請把libgcc_s.so.1放到 /mnt 或其他路徑下，並將該路徑加入LD_LIBRARY_PATH

```
# export LD_LIBRARY_PATH=/mnt:/lib
```

- Step 1.2: 執行 perf report展開剛剛存下的data進行分析

```
# ./perf report
```

將直接在console上顯示各個leaf functions的call stack及執行時間比例

若要直接進行FlameGraph分析，則不一定要進行 perf report

1. 用data產生火焰圖

- Step 2.1: 在EVB上執行 perf script 對 data進行解析，並將輸出結果存檔至pl

```
# ./perf script -i perf.data > p1
```

- Step 2.2: 切换到PC端，至以下連結取得FlameGraph

<https://github.com/brendangregg/FlameGraph>

於PC端執行FlameGraph提供的stackcollapse-perf指令對p1進行符號摺疊並存檔

```
# stackcollapse-perf.pl p1 &> perf.folder
```

- Step 2.3: 於PC端執行FlameGraph提供的pl將perf.folder轉換為svg檔，

生成的svg檔可直接用瀏覽器開啟（建議使用Chrome或Firefox）

```
# flamegraph.pl perf.folder > perf.svg
```

生成火焰图（Flame Graph）命令：

```
./perf record -a -g -F 2599 -p 451 sleep 30  
./perf script -i perf.data > p1  
./stackcollapse-perf.pl p1 &> perf.folder  
./flamegraph.pl perf.folder > perf.svg
```

参考：

<https://www.jianshu.com/p/9a597ff4f5de>

<https://linux.cn/article-4670-1.html>