1 導入

Perf は Linux 用のプロファイラツールです。

1.1 コマンド

perf は git のように `perf <command>`の形式で各種ツールを使用します。 サポートされるコマンドの一覧は `perf` で閲覧できます。

```
  $ perf

  usage: perf [--version] [--help] [OPTIONS] COMMAND [ARGS]

  The most commonly used perf commands are:
    annotate        Read perf.data (created by perf record) and display annotated
code
    archive         Create archive with object files with build-ids found in
perf.data file
    bench           General framework for benchmark suites
    buildid-cache   Manage build-id cache.
    buildid-list    List the buildids in a perf.data file
    c2c             Shared Data C2C/HITM Analyzer.
    config          Get and set variables in a configuration file.
    daemon          Run record sessions on background
    data            Data file related processing
    diff            Read perf.data files and display the differential profile
    evlist          List the event names in a perf.data file
    ftrace          simple wrapper for kernel's ftrace functionality
    inject          Filter to augment the events stream with additional
information
    iostat          Show I/O performance metrics
    kallsyms        Searches running kernel for symbols
    kvm             Tool to trace/measure kvm guest os
    list            List all symbolic event types
    mem             Profile memory accesses
    record          Run a command and record its profile into perf.data
    report          Read perf.data (created by perf record) and display the
profile
    script          Read perf.data (created by perf record) and display trace
output
    stat            Run a command and gather performance counter statistics
    test            Runs sanity tests.
    top             System profiling tool.
    version         display the version of perf binary
    probe           Define new dynamic tracepoints
```

一部のコマンドはカーネルで特殊なサポートを必要とするため使用できない場合があります。各コマンドのオプションの一覧を`-h` で出力することができます。

例:

```
  $ perf stat -h

  Usage: perf stat [<options>] [<command>]
```

```
    -a, --all-cpus        system-wide collection from all CPUs
    -A, --no-aggr         disable CPU count aggregation
    -B, --big-num         print large numbers with thousands' separators
```

1.2 イベント

perf は測定可能なイベントのリストを表示することができます。イベントは複数のソースからなり、一つはコンテキストスイッチやマイナーフォルトなどのカーネルカウンタです。これをソフトウェアイベントと呼びます。

もう一つは Performance Monitoring Unit(PMU)と呼ばれるハードウェアです。PMU はサイクル数、リタイアした命令、L1 キャッシュミスなどのマイクロアーキテクチャイベントを測定するためのイベントリストを提供します。これらのイベントをハードウェアイベントと呼びます。

さらに perf_events インターフェースは一般的なハードウェアイベントの小さなセットも提供します。各プロセッサにおいて

1.2.1 ハードウェアイベント

2 stat によるカウント

2.1 イベント選択オプション

2.2 環境選択オプション

2.3 出力管理オプション

3 record によるサンプリング

3.1 イベントベースサンプリングの概要

3.1.1 デフォルトイベント: サイクルカウント

3.1.2 Period と rate

3.2 サンプルの収集

3.3 プロセッサワイドモード

3.4 Flame Graph

3.5 Firefox Profiler

4 report によるサンプルの解析

4.1 出力制御オプション

4.2 カーネルレポート制御オプション

4.3 プロセッサワイドモード

4.4 オーバーヘッド計算

5 annotate によるソースコードレベルの解析

5.1 カーネルコード上で annotate を使う

6 top によるライブ解析

7 bench によるベンチマーク

7.1 sched: スケジューラベンチマーク

7.2 mem: メモリアクセスベンチマーク

7.3 numa: NUMA スケジューリングと MM ベンチマーク

7.4 futex: Futex ストレスベンチマーク

8 トラブルシューティングとチップス

8.1 ファイルオープンの制限

8.1.1 制限を増やす

8.2 build-id によるバイナリの識別

8.2.1 build-id キャッシュ

8.3 アクセス制御

9 その他のシナリオ

9.1 スリープ時間のプロファイル

10 その他のリソース

10.1 Linux ソースコード

Contents