

Perf

Chapter 3

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。

イベントの一覧は

```
$ perf list
```

で閲覧できます。

2 stat によるカウント

perf stat を使用することでプログラム実行時のイベントを集計できます。

```
$ perf stat -e <event>[,<event>]... <command>
```

で command 実行時の event の集計を行えます。

3 record によるサンプリング

perf record を使用することでプロファイル情報を収集して、perf stat よりも細かいソースコード、命令単位レベルで情報を見ることができます。

```
$ perf record [<options>] <command>
```

文法:

option name	description
-g	コールグラフをレコード
-o, --output <file>	出力ファイルを指定

4 report によるサンプルの解析

perf record で収集したサンプルを report コマンドで閲覧します。

```
$ perf report
```

文法:

option name	description
-g, --call-graph	コールグラフを表示
-i, --input <file>	読み込むファイルを指定
--stdio	TUI ではなく標準出力に表示