

PostMark Benchmark

喜連川研究室

さね 喜連川研究室 産学官連携研究員

概要

PostMark について語ります。

1 はじめに

多くの文献で見かける PostMark Benchmark[1] を実行してみたので最初に非常に簡単な報告を行う。PostMark Benchmark は短命でサイズの小さなファイルを多数扱うアプリケーションの性能を評価するベンチマークである。これはインターネットサーバを想定している。

2 PostMark

2.1 PostMark 概要

短命で小さなサイズのファイルを、大量に処理するアプリケーションの性能を測定するためのベンチマーク。これはインターネット上のサーバ (e-mail, netnews, web-based commerce など) を想定している。標準的なベンチマークソフトの 1 個で文献 [2, 3, 4, 5, 6] などの多くの論文で使用されている。

2.2 入手先等

NetApp 社が実装/配布を行っている。ソースコードや詳細情報は以下の URL から入手できる。
http://www.netapp.com/tech_library/3022.html
http://www.netapp.com/tech_library/postmark.html
<http://www.netapp.com/ftp/postmark-1.5.c>

2.3 参考文献の引き方

文献 [2, 3] では、PostMark に対して

J. Katcher, “PostMark: A New File System Benchmark”, Technical Report TR3022. Network Appliance Inc, October 1997.

という引き方をしている。おそらく

http://www.netapp.com/tech_library/3022.html

のことかと思われる。“J. Katcher” のフルネームは、“Jeffrey Katcher”。

2.4 PostMark の動作

PostMark は、以下の動作を “い”, “ろ”, “は” の順に行い性能を報告する。

(い) **初期ファイルの作成** ランダムテキストファイルを多数個作成する。ファイルサイズは指定範囲内でランダム。“postmark-1.5.c” の 689 行目 “void create_file(buffered)” 参照。

(ろ) **“作成 or 削除” と “読込 or 追記” の反復** “ファイルの作成 or 削除” と “ファイルからの読み込み or ファイルへの追記” の組を多数回繰り返し、その性能を測定する。“postmark-1.5.c” の 852 行目 “int run_transactions(buffered)” 参照。

ファイルの作成 or 削除 ファイルの作成/削除を繰り返しその性能評価を行う。作成 or 削除はランダムに選択される。

ファイルの作成では新規にファイルが作成される。ファイルサイズは、指定された範囲の中からランダムに選ばれる。“postmark-1.5.c” の 689 行目 “void create_file(buffered)” 参照。

ファイルの削除では、ランダムに選択されたファイルが削除される。“postmark-1.5.c” の 728 行目 “void delete_file(number)” 参照。

ファイルからの読み込み or ファイルへの追記

ファイルからの読み込み/ファイルへの追記を行う。読込 or 追記はランダムに選択される。ファイルからの読み込みでは、ターゲットとなったファイルを開き、指定されたブロックサイズ毎にファイル先頭からファイル末尾までファイル全体をシーケンシャルに読み込む。ターゲットファイルはランダムに選択される。“postmark-1.5.c” の 744 行目 “void read_file(number,buffered)” 参照。

ファイルへの追記では、ターゲットとなったファイルを開き、現サイズと指定された最大

サイズの間のサイズとなるように追記を行う。ターゲットファイルはランダムに選択される。追記サイズもランダムに選択される。“postmark-1.5.c”の788行目“void append_file(number,buffered)”参照。

(は) **作成ファイルの削除** 作成した全てのファイルを削除する。“postmark-1.5.c”の729行目“void delete_file(number)”参照。

2.5 PostMark のレポート

PostMark は以下の計測結果を報告する。

- 全所要時間
- トランザクション実行に消費された時間 と 平均トランザクション速度 (files/sec)
- 作成された全ファイルの数 と 平均作成速度 (files/sec)
- 初期に作成されたファイル数 と 平均作成速度 (files/sec)
- トランザクション中に作成されたファイル数 と 平均作成速度 (files/sec)
- 全被読込ファイルの数 と 平均速度 (files/sec)
- 全被追記ファイルの数 と 平均速度 (files/sec)
- 全被削除ファイルの数 と 平均速度 (files/sec)
- トランザクション終了後に削除されたファイル数 と 平均削除性能 (files/sec)
- トランザクション中に削除されたファイル数 と 平均削除性能 (files/sec)
- 全ての読み込まれたデータの合計サイズ と 平均入力性能 (bytes/sec)
- 全ての書き込まれたデータの合計サイズ と 平均出力性能 (bytes/sec)

上記の“トランザクション”とは“作成 or 削除 と 読込 or 追記”の反復のことであり、初期ファイルの作成や最後のファイルの削除が含まれていない。

2.6 PostMark の特徴

ファイルシステムの上から負荷をかけて性能を測るためファイルシステムの実装/振る舞いの影響を受ける。例えばファイルシステムのキャッシュの影響をそのまま受ける。よって多くの場合は NFS よりも iSCSI の方が高性能となる。

3 実行方法

3.1 入手 と コンパイル

```
> wget http://www.netapp.com/ftp/postmark-1.5.c
> gcc -O postmark-1.5.c -o postmark
```

3.2 簡易実行

まずは、全て default で実行する方法を紹介する。以下の通りで計測が可能である。下線部が、ユーザによる入力である。

```
[sane@hostname ~/postmark]$ ./postmark
PostMark v1.5 : 3/27/01
pm>show
Current configuration is:
The base number of files is 500
: (SNIP!)
Report format is verbose.
pm>run
Creating files...Done
: (SNIP!)
Files:
764 created (764 per second)
: (SNIP!)
pm>quit
[sane@hostname ~/postmark]$
```

3.3 実行の流れ

以下に、PostMark の典型的な実行手順を紹介する。

1. PostMark の起動
2. ベンチマークの各種パラメータの設定
3. ベンチマークパラメータの確認
4. ベンチマークの実行
5. PostMark の終了

各手順について以下に説明を行う。

```
[sane@hostname ~/postmark]$ ./postmark
PostMark v1.5 : 3/27/01
pm>
```

図 1: PostMark の起動

(1) **起動** 以下の様に PostMark を起動する.

```
./postmark
```

これにより PostMark のプロンプトが表示される.
図 1 参照.

```
./postmark コマンドファイル
```

としてコマンドが記述されているファイルを指定し, 実行の自動化を行うことも可能である. 第 ??? で後述する.

(2) **設定** 工事中である

(3) **確認** 工事中である

(4) **実行** 工事中である

(5) **終了** 工事中である

4 ベンチマークの設定

PostMark では 10 種類のベンチマークパラメータを設定することが可能である. 全てのパラメータは PostMark のコマンドプロンプトから

```
pm>set パラメータ名 値
```

と指定することができる. 以下に, その全て (size, number, seed, transactions, location, read, write, buffering, bias read, bias create) の解説を記す.

4.1 size

ベンチマークにおいて処理される (作成される) ファイルのサイズの上限/下限を指定する. 指定方法は,

```
set size ファイルサイズ下限 ファイルサイズ上限
```

と, 2 個の値を指定して行う.

```
set size ファイルサイズ
```

と, 値を 1 個だけ指定した場合はファイルサイズは指定された値のみ許されることとなる. 例えば,

```
set size 10000 20000
```

と指定すれば, ベンチマークで処理されるファイルのサイズは 10,000 バイトから 20,000 バイトの範囲となる.

5 実行例

```
[sane@hostname ~/postmark]$ ./postmark
PostMark v1.5 : 3/27/01
```

```
pm>show
```

Current configuration is:

The base number of files is 500

Transactions: 500

Files range between 500 bytes and 9.77 kilobytes in size

Working directory: /home/sane/postmark

Block sizes are: read=512 bytes, write=512 bytes

Biases are: read/append=5, create/delete=5

Using Unix buffered file I/O

Random number generator seed is 42

Report format is verbose.

```
pm>run
```

Creating files...Done

Performing transactions.....Done

Deleting files...Done

Time:

1 seconds total

1 seconds of transactions (500 per second)

Files:

764 created (764 per second)

Creation alone: 500 files (500 per second)

Mixed with transactions: 264 files (264 per second)

243 read (243 per second)

257 appended (257 per second)

764 deleted (764 per second)

Deletion alone: 528 files (528 per second)

Mixed with transactions: 236 files (236 per second)

Data:

1.36 megabytes read (1.36 megabytes per second)

4.45 megabytes written (4.45 megabytes per second)

```
pm>quit
```

```
[sane@hostname ~/postmark]$
```

参考文献

- [1] Jeffrey Katcher. Postmark: A new file system benchmark, October 1997.
- [2] Wee Teck Ng et al. “Performance Evaluation and Improving of Sequential Storage Access using iSCSI Protocol in Long-delayed High throughput Network”. In *Proc. of IEICE The 14th Data Engineering Workshop*, March 2003.
- [3] Prasenjit Sarkar, Sandeep Uttamchandani, and Kaladhar Voruganti. “Storage over IP: When Does Hardware Support help?”. In *Proc. FAST 2003, USENIX Conference on File and Storage Technologies*, March 2003.
- [4] Peter Radkov, Li Yin, Pawan Goyal, Prasenjit Sarkar, and Prashant Shenoy. “A performance Comparison of NFS and iSCSI for IP-Networked Storage”. In *Proc. FAST 2004, USENIX Conference on File and Storage Technologies*, March 2004.
- [5] 藤田智成 小河原成哲. “iSCSI ターゲットソフトウェアの解析”. In 先進的計算基盤システムシンポジウム *SAC SIS 2004*, May 2004.
- [6] Inc. TechnoMages. Performance comparison of iscsi and nfs ip storage protocols.
http://www.technomagesinc.com/papers/ip_paper.html
.