

Japan OpenSolaris Users Group

opensolaris™

LLプログラマの為のD-Trace(PHP編)

~PHPプログラマなら、1度はSolarisで動かしてみろ!~

Ver 0.2

OpenSolaris Users Group Leader

ジャストプレイヤー株式会社

瀧 康史 / TAKI, Yasushi¹

www.opensolaris.org

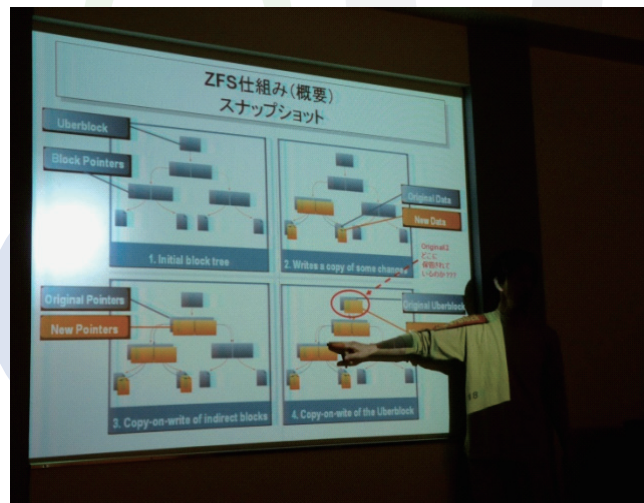
自己紹介

- 瀧 康史(たき やすし)
- ジャストプレイヤー株式会社 代表取締役社長
- 出身地
 - 静岡市(旧清水市)
- 業務履歴
 - ライター業(テクニカルライター)
 - S社‘X紙’(S社のパソコン向け雑誌)を中心に5年間。
 - S社にてS社用ゲーム機のゲームソフトを5年間制作。
 - ゲームディレクター、アシスタントプロデューサー、プランナーなどの雑用(笑) 係
- 2001年、有限会社ジャストプレイヤーを起業
 - 現: ジャストプレイヤー株式会社



日本 OpenSolaris ユーザグループ

- 日本のOpenSolarisのコミュニティです。
- Oracle SolarisからOpenSolaris派生のコミュニティベースのディストリビューション「OpenIndiana、Nexenta等」まで、すべてを扱っています。
- メーリングリスト(ug-jposug@opensolaris.org)で活発に意見交換されています。
- 月に一度、最終土曜日に、「東京OpenSolaris勉強会」を開催しています。
- IRC(irc://irc.freenode.net/opensolaris-jp)あります。
- 初心者大歓迎です。



なぜ、PHP を Solaris で?

● SolarisのPHPのメリット

- 動いてるサーバの状態を見やすい。
- メモリ、プロセス、ボトルネック等々。

● Solaris 11やOpenIndianaでは、インストールも普通にできます

● 本番機をSolarisにするメリット

- SMF
- BART
- Project/TASK
- などなど……

```
web/php-52 5.2.12-0.151.0.1 既知 ----
web/php-52/documentation 5.2.12-0.151.0.1 既知 ----
web/php-52/extension/php-apc 3.0.19-0.151.0.1 既知 ----
web/php-52/extension/php-fileinfo (localhost) 1.0.4-0.151 既知 ----
web/php-52/extension/php-fileinfo/src (localhost) 1.0.4-0.151 既知 ---
----
web/php-52/extension/php-idn 0.2.0-0.151.0.1 既知 ----
web/php-52/extension/php-memcache 2.2.5-0.151.0.1 既知 ----
web/php-52/extension/php-mysql 5.2.12-0.151.0.1 既知 ----
web/php-52/extension/php-pear 5.2.12-0.151.0.1 既知 ----
web/php-52/extension/php-pgsql 5.2.12-0.146 既知 --o--
web/php-52/extension/php-pgsql90 (localhost) 5.2.12-0.151 既知 ----
web/php-52/extension/php-pgsql90/src (localhost) 5.2.12-0.151 既知
----
web/php-52/extension/php-suhosin 0.9.29-0.151.0.1 既知 ----
web/php-52/extension/php-tcpwrap 1.1.3-0.151.0.1 既知 ----
web/php-52/extension/php-xdebug 2.0.5-0.151.0.1 既知 ----
web/server/apache-22/module/apache-php5 5.2.12-0.151.0.1 既知
```

```
pkg install -v web/php-52
```

などで、インストール可能です。

ディレクトリ
/etc/apache2/2.2/
/var/apache2/2.2/
/usr/apache2/2.2/

/etc/php/5.2/

等々。

php をどう動かしていますか？

- mod_php
 - Preforkしたapache上で動く (workerでは事実上動かない)
 - メリット: Apacheからリクエストが同一プロセス上で飛ぶので、速い。
 - デメリット: メモリ効率がめちゃめちゃ悪い
- CGI/FastCGI
 - 呼び出しは遅い (FastCGIなら許せるかなあ?)
 - メモリ効率は良い。
- 自分でFork…
 - (略)

メモリ効率の悪さ

- ForkしたApacheのインスタンスが個別に確保するメモリ
- 一度確保したら、プロセスが死ぬまで解放されないメモリ
 - MaxRequestPerChildの回数でForkしたプロセスが死ぬ時 (デフォルト無限)
 - プロセス数がシュリンクする時に減る (最近のapacheは割と減らすのは早い)
 - 宇宙線などがCPUのいいところにちょうど当たって死ぬ
 - 等

実際の状態を見てみましょう!

デモ

- prstat -s size
- prstat -s swap

pkg(4)

User Commands

NAME

pkg - image packaging retrieval client

SYNOPSIS

/usr/bin/pkg [options] command [cmd_options] [operands]
/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...
/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...



テーマ：WEB サイトの高速化

せっかく作ったphpプログラムが遅くて我慢できない。

- 説明略

遅くなってしまった例のサイトを見てみましょう。

大人の事情（現場）

- 大人の事情で、担当者が変わって接ぎ木されたプログラム
- 例
 - ある値を取得する関数があり、それはファイルを開いて得る物だった。
 - ある値を保存する関数があり、それはファイルに開いて保存する物だった。
 - その後……
 - この関数便利じゃね?→至る所で流用
 - カプセル化は悪いことじゃないけれども、そのクラスがどのくらい重いとか、どのくらいインスタンスのメモリを食うか、あんまり意識できないよね?
- こういうのが増えていくと、どこにボトルネックがあるかわからない??

安易な逃げ

- その1) そうだ!マシンを速くしよう!
- その2) そうだ!OSの設定のチューニングでなんとかしよう!

OS で、チューニングできる余地？

基本はsyscallの「中」の時間しか、短縮できません。

syscallの使われ具合を調べる

- prstat -mL
- syscallの値をみる。
- truss -c
- iostat

usrが大きい場合は、

- 速いCPUを用意する
- プログラムを高速化する



やっば、プログラムじゃね？

どこを治すのかを調べる必要があるが、小手先で治しても効率わるいよね。

- なぜ、遅いのか調べる
- どこが、遅いのか調べる
- 遅いところがチューニングできるのか？
- 遅いところは、何度も呼ばれてるのか？
- 遅いロジックの修正をする。

やっば、プログラムじゃね？

今回は、D-Trace ToolKitを使ってみましょう

インストール方法

```
pkg install -v dtrace/toolkit
```

NAME

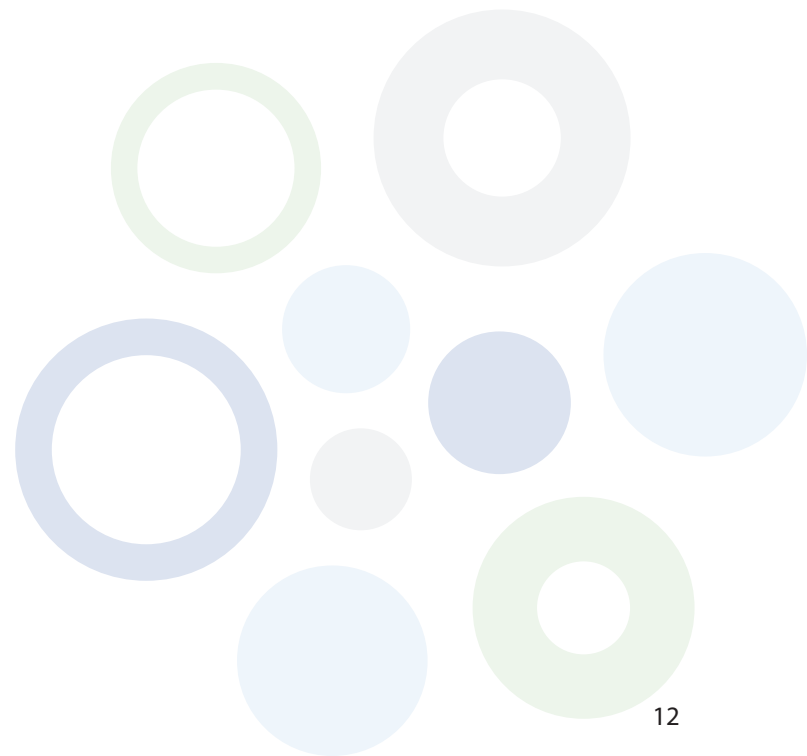
pkg - image packaging retrieval client

SYNOPSIS

/usr/bin/pkg [options] command [cmd_options] [operands]

/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...

/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...



関数毎のあたりを調べる

- php_calltime/php_calldist/php_funccall/php_cputime/php_cpudist
 - php関数毎の処理時間等を測定する。php_calltime.dが、スーパーセット。exclusive/inclusive/calltimeから比較可能。
 - dist系は表示違い。本番機では表示が多すぎて見るのがむしろ大変。
 - php_cpu*は、CPU処理時間だけを計測。
 - php_funccallは呼び出し回数だけ。

関数の呼び出しの流れを調べる

- php_flow/php_flowinfo/php_flowtime
 - 現在実行されているphp関数のsnoop (本番機でやるととても重いので注意)
 - flowでは関数が、flowinfoでは呼ばれていく順が、flowtimeでは処理時間が表示

NAME

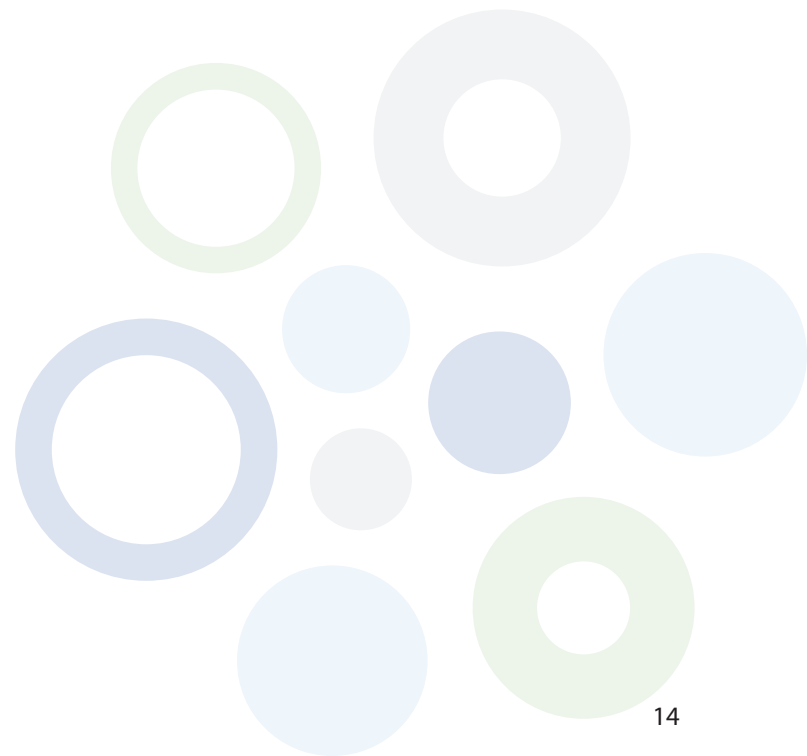
pkg - image packaging retrieval client

SYNOPSIS

/usr/bin/pkg [options] command [cmd_options] [operands]

/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...

/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...



重いリソース使ってないか？

- php_malloc/php_syscall/php_syscolor
 - libc経由のmalloc状況や、syscallの発行状況を見る。

NAME

pkg - image packaging retrieval client

SYNOPSIS

/usr/bin/pkg [options] command [cmd_options] [operands]

/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...

/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...



参考) 重い VirtualHost は?

- php_who
 - 実行されているPID、UID、ファイルを見る (どのVirtualHost?)

NAME

pkg - image packaging retrieval client

SYNOPSIS

/usr/bin/pkg [options] command [cmd_options] [operands]

/usr/bin/pkg install [-nvq] [-no-refresh] [-no-index] pkg_fmri...

/usr/bin/pkg uninstall [-nvq] [-no-index] pkg_fmri...



ご静聴有り難うございました。

ご静聴、ありがとうございました

OpenSolaris Users Group

瀧 康史

