□□ CMS News Letter □□□□□□□□□□□□□Core Micro Systems □□□

本日発表「Prime STOR ZFS ENTERPRISE」
-オンデマンドでの性能と拡張性を追求した理想的な仮想環境下のストレージ-

□□□□□□□□□□□□□□□□□□□□□□□□□ Vol.010 2010/10/01 □□

いつもお世話になっております。
コアマイクロシステムズの吉政でございます。

今号は本日報道発表いたしました「Prime STOR ZFS ENTERPRISE」をご紹介したくメール致しました。

「Prime STOR ZFS ENTERPRISE」はNexentaStor搭載の弊社フラグシップ製品の一つであり、昨今の仮想環境で求められるオンデマンドでの性能と拡張性を追求した製品になっています。是非ご注目頂きたいです。

それでは、今号も宜しくお願いいたします！！！

□□ INDEX □□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□

□□

◆◆>[News & Topics]<◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

◇9月30日報道発表予定

強力なストレージ仮想化機能により、オンデマンドでの性能強化と
柔軟な容量拡張を実現した
「Prime STOR ZFS ENTERPRISE」を10月1日より発売開始

解説：近年急速に進むサーバ仮想化とストレージ効率化のニーズに対応する革新のストレージシステム「Prime STOR ZFS ENTERPRISE」を発表しました。
本ストレージシステムは、強力なストレージ仮想化機能により、オンデマンドでの性能強化と容量拡張に柔軟に対応し、仮想統合環境や各種のエンタープライズアプリケーションの将来的なストレージニーズに的確に応えます。
NexentaStor搭載のフラグシップになります。是非ご注目ください。（吉政）

報道発表原文：http://www.cmsinc.co.jp/news/2010/news2010093001.html

◆◆>[お知らせ]<◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆◆

◇　事例強力のお願い　◇

弊社ではお客様事例を募集しています。
ご協力をいただける可能性があるお客様は是非下記をご覧いただき、お申しつけ頂ければ幸いです。

＜対象製品＞
・弊社全製品

＜公開方法＞
A)社名のみの公開
B)簡単な事例公開　※下記URLの事例を参考にしてください。
　http://www.cmsinc.co.jp/casestudy/casestudy_04.html
C)報道発表による事例公開

※上記「C)」についてはニュース性が重要であり、ご導入から1年以内の事例に限ります。

＜取材方法＞
・弊社よりインタビューワーとカメラマンを派遣し、取材を致します。
・公開用の資料は全て弊社側で用意します。

＜補足＞
・事例公開にあたっては貴社の許可を得たコンテンツのみ公開いたします。
・貴社からの掲載取りやめ依頼があった場合は直ちに掲載を中止します。

＜お申し込みと質問＞
本メールマガジンの返信にてお申し付けください。

◇　ドキュメントセンターのご紹介　◇

弊社のドキュメントセンターをご覧になったことはありますか？
去年の10月に開設して以来、技術資料などのダウンロードを大変多くの方にご利用頂いています。もし、まだ利用された事が無い方がいらっしゃいましたら、この機会に是非お試しください。ストレージのニッチな使い方や数値データなど役立つ情報が満載です。

＜ドキュメントセンター＞
http://www.cmsinc.co.jp/doccenter/index.html

◆◆>[テクニカルコラム：かんたん♪　ＺＦＳ　]<◆◆◆◆◆◆◆◆◆◆◆◆

第十回：DTrace

◇◇

こんにちは。今年の夏は記録的な暑さでしたが、お彼岸を過ぎ、ようやく涼しくなってきました。みなさんいかがお過ごしでしょうか。このコラムでは実際に ZFS を使う上で役に立つ Tips を紹介します。

今までのファイルシステムとは一味違う ZFS　を活用して面倒なファイル管理の苦労を減らしましょう！

パフォーマンス改善を行う時、負荷に対して処理能力を持て余している部分を補強してもほとんど改善は望めません。一方、処理能力を目一杯使っている部分を補強すると、大きな改善を期待することができます。

パフォーマンスを改善するには、処理状況をモニタし、システムの最も弱い部分を発見するのが、最初の一歩となります。前回は iostat、mpstat、vmstat を利用し、I/O、プロセッサ、メモリリソースの利用状況を確認する方法を説明しました。今回は DTrace という機能について説明します。

DTrace はサンマイクロシステムズが開発した、システムの動的なトレースを行うための機能です。複雑な機能ですので、完全に使いこなすには、時間をかけてドキュメントを読む必要があります。しかし、典型的なプログラムを天下り的に使用して、ボトルネックの調査に役立てることもできます。

まずは簡単な DTrace プログラムを紹介します。

```
#!/usr/sbin/dtrace -s

#pragma D option quiet

io:::start
{
    @[args[1]->dev_statname] = quantize(args[0]->b_bcount);
}
```

テキストエディタでこのプログラムが書かれたファイルを作成し、以下のように実行してください（ファイル名を iosize.d）とします。

```
# chmod 700 iosize.d
# ./iosize.d
^C

  sd1
   value  ------------- Distribution ------------- count
      16 |                                         0
      32 |                                         2
      64 |                                         0
     128 |                                         0
     256 |                                         0
     512 |                                         34
    1024 |                                         42
    2048 |                                         41
    4096 |                                         17
    8192 |                                         10
   16384 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 53447
   32768 |                                         2
   65536 |                                         4
  131072 |                                         1
  262144 |                                         0
```

自動的に終了しないので Ctrl + C を押して終了させます。

このプログラムは、I/O ブロックサイズの統計情報を表示します。この例では 16384 [byte] = 16 [Kbyte] のアクセスが 53447 と表示されています。プログラム実行中に I/O 活動が無かった場合は何も表示されません。

例えば、バックグラウンドでディスクを読み込むタスクを実行させてから、iosize.d を実行してみましょう。/dev/rdsk/c0t0d0p0 の部分は、お使いのシステムに合わせて変更してください。

```
# dd if=/dev/rdsk/c0t0d0p0 of=/dev/null bs=32k &
[1] 26713
# ./iosize.d
^C

  sd1
   value  ------------- Distribution ------------- count
     256 |                                         0
     512 |                                         1
    1024 |                                         0
    2048 |                                         0
    4096 |                                         2
    8192 |                                         4
   16384 |                                         0
   32768 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 11214
   65536 |                                         0
```

当然 32768 [byte] = 32 [Kbyte] の読み込みが多くカウントされます。DTrace プログラムに戻って解説すると、io:::start の部分はプローブと呼ばれます。 プローブは provider:module:function:name という形式で表現されます。

```
io:::start
{
    @[args[1]->dev_statname] = quantize(args[0]->b_bcount);
}
```

このプログラムは io プロバイダの start というプローブを使用して統計情報を取得します。このプローブについては『Solaris 動的トレースガイド』において、次のように説明されています。

「周辺機器（デバイス）または NFS サーバーに対する入出力要求が発行される直前に起動するプローブ。args[0] は、入出力要求の bufinfo_t をポイントしています。args[1] は、入出力要求の受け取り側デバイスの devinfo_t をポイントしています。 args[2] は入出力要求に対応するファイルの fileinfo_t をポイントしています」

bufinfo_t 構造体は次のように定義されています。

```
typedef struct bufinfo {
    int        b_flags;      /* flags */
>>> size_t     b_bcount;     /* number of bytes */
    caddr_t    b_addr;       /* buffer address */
    uint64_t   b_blkno;      /* expanded block # on device */
    uint64_t   b_lblkno;     /* block # on device */
    size_t     b_resid;      /* # of bytes not transferred */
    size_t     b_bufsize;    /* size of allocated buffer */
    caddr_t    b_iodone;     /* I/O completion routine */
    dev_t      b_edev;       /* extended device */
} bufinfo_t;
```

devinfo_t 構造体は次のように定義されています。

```
typedef struct devinfo {
    int        dev_major;    /* major number */
    int        dev_minor;    /* minor number */
    int        dev_instance; /* instance number */
    string     dev_name;     /* name of device */
>>> string     dev_statname; /* name of device
                                 + instance/minor */
    string     dev_pathname; /* pathname of device */
} devinfo_t;
```

これらはシステムの I/O 入出力関数に渡された引数です。io:::start プローブを使うとこららのパラメータを参照することができます。
@[] = quantize() は集積体とよばれ、 指定された式の二乗分布を作成します。 集積関数には quantize() 以外にも、 count() - 呼び出された回数、sum() - 指定された式の合計、などがあります。

サンプルプログラムでは、 args[0]->b_bcount （ブロックサイズ）の二乗分布を args[1]->dev_statname （デバイス名）毎に集計します。

```
@[args[1]->dev_statname] = quantize(args[0]->b_bcount);
```

このように、プローブがどのような契機で、どのような値を取得することができるのかを知っていれば、 iostat や mpstat よりも実際の処理と密接に結びついた情報を得ることができます。

今回は DTrace の基本的な部分を紹介させていただきました。次回も引き続き、いつくかの DTrace サンプルプログラムを使用した、システム情報の収集方法について紹介します。

（つづく）

■ 参考文献

(1) サンマイクロシステムズ、"SunOS リファレンスマニュアル 1 : ユーザコマンド"、Part No. 819-1210-13、2007 年 7 月

(2) サンマイクロシステムズ、"SunOS リファレンスマニュアル 1M : システム管理コマンド"、Part No. 819-1211-13、2007 年 7 月

(3) サンマイクロシステムズ、"Solaris 動的トレースガイド", Part No. 819-0395-12、2008 年 10 月

(4) サンマイクロシステムズ、"Dtrace ユーザガイド", Part No. 819-6259-10、2006 年 5 月

（高田　浩生）

□□
※　CMS BizStor Updateのバックナンバーはパートナーログインにアップされています。

編｜集｜後｜記｜

以前もお話しましたが、私は今年の4月より独立し、マーケテイングコンサルティング会社を経営しています。コアマイクロシステムズ様とはアドバイザー契約をしおり、その中で、メールマガジンの発行やWebサイトの更新、事例取材を担当しております。事例としてご協力を頂ける場合はお気軽にお知らせ下さい。ご挨拶を兼ねて取材にお伺い致します。何卒宜しくお願いいたします。
（吉政）

コアマイクロシステムズでは、お客様事例を募集しております。事例公開にご協力いただきましたお客様の社名やお取り組みを弊社のホームページや販促資料に記載し、皆様のビジネスの露出度向上に些少でも貢献出来ればと思います。お気軽に弊社担当までお申し付けください。

―◇本メールマガジンの登録解除方法◇―――――――――――――――――――
このメールに対する登録解除は★そのまま返信★でその旨をお伝えください。

本メールマガジンの送付先追加と送付先変更方法
このメールマガジンに★そのまま返信★で下記の項目をCut & Pasteして項目を埋め、返信して下さい。

変更の場合、旧送付先：
（１）メールアドレス：
（２）氏名　　：
（３）会社名　　：
（４）部署名　　：
（５）役職名　　：

発行人：コアマイクロシステムズ株式会社　吉政　忠志