2011 年 8 月 1 日
ぷらっとホーム株式会社
技術部発行

フラッシュメモリーのシステム利用手引き

今後、ハードディスクの代替えストレージとして期待されるフラッシュメモリーですが、システムを安定運用させるためにはフラッシュメモリー特有の性質を理解し、それに合わせた運用方法を用いる必要があります。
本書では、サーバストレージとして SSD や CF などを利用するために有効な手法を解説します。

性質 1. データの書き換え可能回数が少ない

ハードディスクの一般性能では、同じブロックアドレスに対して、約 100 万回のデータ書き換えが可能です。
フラッシュメモリーの場合は、タイプにより異なります。

SLC タイプ（シングルレベルセル）
データが保存される 1 つのセルに対してデータが 1 ビットのみ保存されます。（0 か 1 の 2 値）
セルが単純なので、書き換え可能回数が数万回～数十万回と、ハードディスクより 1～2 桁程度落ちますが、ウェアレベリング（均等処理）の技術を組み合わせて利用した場合、理論的にハードディスクと同等の書き換え回数を達成可能です。（後述）

MLC タイプ（マルチレベルセル）
データが保存される 1 つのセルに対してデータが複数ビット保存されます。（2 ビット保存の場合で 00・01・10・11 の 4 値、3 ビットなら 8 値）
データが 0 か 1 かの単純な SLC に比べると、MLC は保存する値にレベル差をつけて保存しますが、書き換えを繰り返す内に保存するデータのレベルにバラ付きが出てしまうため、その保存データの信頼性が SLC よりもさらに 2 桁ほど落ちてしまいます。（2 ビットの場合で 2 桁）
一般的に、数百回～数千回程度の書き換え性能となりますが、近年、より高容量を安価に達成するために設計スケールが微細化しており、その影響で書き換え可能回数は減る一方と言えます。
一般的に安価な MLC 製品の書き換え可能数は 200 回程度であることを前提にすべきと言えます。

単純に区別すると SLC は書き換え回数が数万回以上を達成するが、MLC は 2 倍以上のデータを保存できる代わりに書き換え可能回数を数百回程度の 2 桁分落としていると言う事になります。

性質 2. ウエアレベリングで理論上の書き換え可能回数を増やしている

ハードディスクの場合、同じブロックアドレスに対して書き換え命令を発行すると、物理的に同じ場所へデータが書き込まれます。（代替処理は例外です。）

フラッシュメモリーで、これと同じようにデータを書いてしまうと、書き換え可能回数が少ないので、ハードディスクと同じ様なストレージ寿命を維持できなくなります。これを回避する技術がウエアレベリングと言います。

例えば、ファイルの更新情報などが保存されているブロックは、リードアクセスでもそのアクセス日時などが頻繁に更新されるので、ウエアレベリングが無いと特定のブロックがあっという間に書き換え可能回数の上限に達成してしまいます。
ウエアレベリングがある場合は、それぞれの物理ブロックの書き換え回数をカウントしているので、コンピュータからの書き込み命令がくると、物理ブロックの中で一番書き換え回数の少ないブロックを選択して書き込み、その物理ブロックに論理ブロックのアドレスをふります。
こう言った仕組みによって、同じ物理アドレスに対してデータの書き込みが何度も発生せず均一化されます。

しかし、古い方式のウエアレベリングには一つの問題がありました。
書き換えの必要の無いデータが、フラッシュメモリー内の大半を占める場合、ウエアレベリングの効果が発揮できないという問題です。
書き換えされないデータとは、例えば OS のシステム領域ですが、OS インストール後、OS のトラブルなど必要のない限りデータの書き換えはしません。

もし、書き換えられないデータが 8GB のフラッシュメモリーのうち 7GB に置かれていたらどうなるでしょうか？
つまり、ウエアレベリングされるブロックというのは、残った 1 GB の部分だけになります。
7 GB の書き換えされないデータが置かれたブロックは、データを書き換えた回数として 1 回のまま一生そのまま置かれることになります。
一方、何度も書き換えられる 1 GB の領域は、短期間に書き換え可能数を消費し老化していく事になります。

そこで、現在ではスタティックウエアレベリングと呼ばれる方式が使われるようになりました。（古い方式のウエアレベリングは、区別のためダイナミックウエアレベリングと呼ばれています。）スタティックウエアレベリングでは、書き換えられない領域も含めて均一化するように、書き換えられないデータを別のブロックへ時々移動します。（書き換え回数にある程度の開きが生じた時に老化しているブロックへ移動。）
つまり、この移動によって、特定ブロックの老化集中を防いでいるわけです。

結果、前述した 1GB:7GB の使用例では、論理的な書き換え可能回数が実質 8 倍となるわけです。

性質 3. ECC エラーが発見されてもギリギリまで書き換えない

ストレージに保存されるデータは、必ずパーフェクトな状態で保存されているわけではなく、ある程度のレートでデータのビット落ちなどが発生してしまいます。
理想的には、このエラーが発見されたら即データを正常状態に回復して、ストレージに保存しなおすべきなのですが、フラッシュメモリーには書き換え可能回数の上限が低いという問題もあり、エラーが発見される度に書き換えていてはメモリーの老化を促進してしまいます。

このため、ECC エラーのビット数がある程度の数になるまでは、こう言ったデータ回復・保存しなおしのプロセスを起動しないように設計されています。
しかし、この機能はデータ保存の信頼性とはトレードオフされてしまいます。

そのエラーのあるブロックを次の機会に読み込んだ時、さらにエラービットが増えていたら、場合によってはエラー回復も不可能になってしまいます。
この辺のリスクがトレードオフになるわけです。

性質 4. 実際の書き換えブロックサイズがハードディスクよりも大きい

ハードディスクは 512 バイトを 1 ブロックとして管理しています。
コンピュータは、データをこのサイズに合わせて読み書きしています。
しかし、フラッシュメモリーの場合、512 バイトという小さなブロックサイズでは、メモリー容量を十分に確保できないために、実際の物理ブロックサイズは 16KB や 32KB・64KB と大きなサイズで管理されています。
ただし、ハードディスクとのシステム互換を取るために、コンピュータからの見た目は 1 ブロック 512 バイトとして利用出来るようにエミュレーションされています。

ここで注意すべき問題点とは、物理ブロックサイズが 32KB の場合、512 バイトのデータを書き換えただけでも、32KB のデータを書き換えたのと同じ扱いになる点です。

フラッシュメモリーはウエアレベリングするため、耐久性能を書き換え回数で表現せず、最大書き込み可能容量で表現されることが多くあります。
例えば、200 回書き換えが可能な 8GB フラッシュメモリーでは、最大で書き込めるデータ量が 8GB×200＝1600GB となるわけです。

この 1600GB から前述の 512 バイトのファイルを書き換えることを前提とした場合、よく勘違いされるのは 1600GB ÷ 512 で約 31 億回の書き換えが可能と解釈されてしまいますが、実際には 1600GB ÷ 32K となるので 5000 万回が正しい計算となります。
こう言った部分でシステム設計時に予定の耐久年数を経ずにフラッシュメモリーの寿命が来てしまった事例があります。

フラッシュメモリーをハードディスクの代替え利用するには、以上の性質を理解した上でシステム設計する必要があります。
その他にも、安定利用のためのノウハウがあります。

ノウハウ 1. 使い始める前に全面ゼロ書きを行う。

（当社にてＣＦにソフトウェアを組込み販売しているモデル・シリーズについては当社組込み時に全面ゼロ書きを実施しておりますので、お客様での実施はしないでください。）

SSD や CF などのフラッシュメモリーは、工場で生産されてから高温にさらされたり、飛行機に乗せられ低温にさらされたり、そして長期在庫されたりしながら利用者の手元に届く事になります。

工場でフォーマットされても、そういった様々な環境を経てくると、メモリーセルのデータにもビット落ちが発生する可能性があります。
このビット落ちのある状態で使い始めるのは、高いリスクがあります。

システムのセットアップだけでは、すべてのメモリーセルがリフレッシュされるわけではないので、ビット落ちのしたセルを残したままですと、運用開始後に回復できない ECC エラーが検出される場合があります。

これが発生してしまうと、システムの運用状態によっては、ECC エラーを回復できないブロックが永久に残り続けるケースも有り得ます。

この問題には dd コマンドなどで、全データ領域を 0 でクリアすることで回避可能です。
例) dd if=/dev/zero of=/dev/sda1 bs=1M

ノウハウ 2. やたら書き換えの多いテンポラリーファイルは RAM ディスクに置く

特に不揮発性である必要がなく、やたら書き換えの多いデータファイルなどは RAM ディスクに配置すると、フラッシュメモリーの老化を抑えることができます。

例えば DHCP サーバーの場合、リースファイルがそういったファイルです。
一定時間毎に IP アドレスの貸出情報ファイルが更新されますが、こう言ったファイルを RAM ディスクに配置するだけでも、システムは大幅に冗長性が高くなります。

例) mount　-t tmpfs –o size=32m ram /var/ramdisk

この例では、32MB の RAM ディスクを/var/ramdisk に確保していますが、このディレクトリ以下に書き換えが頻繁なファイルを配置することで対策となります。

ノウハウ 3. アクセス日時の更新を行わない

前述の通り、ファイルの更新情報などが保存されているブロックは、リードアクセスでもそのアクセス日時などが頻繁に更新されますが、このアクセス日時の更新を行わないことにより、書き換えの回数を減らすことが可能です。CF マウント時のオプションに「noatime」を与えることで、アクセス日時の更新を行わなくなります。

例) mount –o noatime /dev/sda1 /mnt

ただしこの方法を有効にした場合、アクセス日時を参照するアプリケーション(メール関連やファイルバックアップ等)の動作に問題が出る場合がありますので、良く注意して利用して下さい。