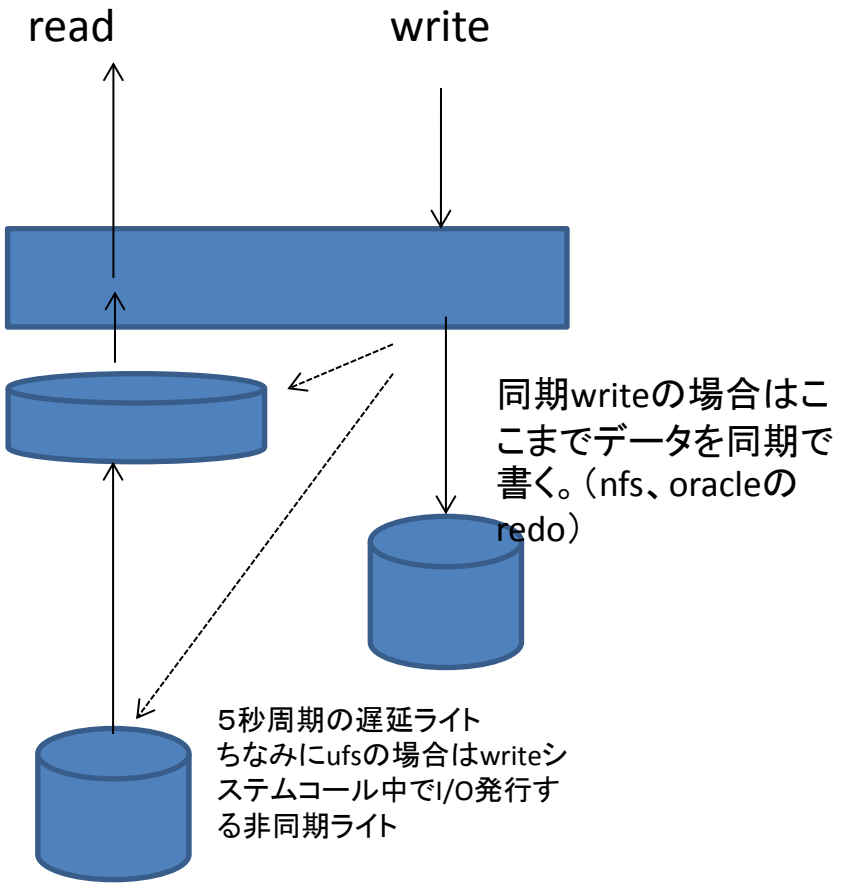
zfsのarcは面倒だ！

zfsのキャッシュ階層



zrcキャッシュ

- カーネルメモリ扱で確保される
  - ufsではファイルシステムキャッシュはfreeメモリ扱だった
- mdbの::arcもしくは kstat –n arcstatsで使用量がレポートされる
- どんどんに大きくなる(ただしzfs_arc_maxが上限)
- 秒１回大きさがチェックされて、freeメモリ量がlotsfree（=主記憶/64） + desfree(=主記憶/128) + user_reserve_hint_pct分 より小さくなると、２秒に１回の割で縮小する。

zrcキャッシュの使用量(論理値)

```
::arc
 (中略)
c                               =       1522 MB
c_min                           =        190 MB
c_max                           =       1522 MB
size                            =        589 MB
hdr_size                        =    3435808
data_size                       =  576789504
other_size                      =   37964144
arc_meta_used                   =        125 MB
arc_meta_limit                  =        380 MB
arc_meta_max                    =        125 MB
arc_no_grow                     =          0
arc_tempreserve                 =          0 MB
```

zrcキャッシュの使用量(物理量)

```
# mdb -k
> ::memstat
```

Page Summary	Pages	MB	%Tot
Kernel	**131827**	**214**	**11%**
ZFS File Data	**114921**	**748**	**37%**
Anon	73819	288	14%
Exec and libs	3680	14	1%
Page cache	24680	96	5%
Free (cachelist)	8215	32	2%
Free (freelist)	162408	634	31%
Total	519550	2029	
Physical	519549	2029	

zfs arc への不満

- freeメモリを圧迫し、一時的なアプリケーションの起動エラーや遅延などを発生するので、arcサイズをシステム管理者がチューニング必要
- ufsのときは、solaris7あたりからufsキャッシュがfreeメモリ扱になったのでチューニング不要で楽なOSだった。
- solaris 2.6まではpageout頻発してシステムハングするため、lotsfreeのチューニング必要だった。昔に逆戻り！

zrcキャッシュの問題点

- 縮小時のカーネルメモリのガベージコレクションを引き起こす
  - x86だとページテーブルの余裕分なども破棄され、プロセス生成に遅延を起す。
  - 高速なディスクを使っている場合に、カーネルメモリ解放が追い付かず、数秒ー10数秒カーネルがハング
  - O社推奨のuser_reserve_hint_pctでは、メモリ枯渇状態を発生させてarcの増大を防ぐため、ハングを回避できない。
  - 結局はシステムごとにzfs_arc_maxパラメタを小さく設定してarc キャッシュの量に適切な上限設定するしかない。

zfc_arc_maxパラメタ

- /etc/systemで設定。
  - 変更後リブート必要のため嫌われている
- arcにキャッシュされる有効データ量の上限値
  - デフォルトは主記憶全部（ヤバイ！）
  - solaris11.2までは、データ部とメタ部の和
  - solaris11.3では データ部のみ
  - 論理量で指定するので、使用される物理ページ量はガベージコレクト(kernelメモリアロケータのreap)されるまで肥大化する。これがmdb ::memstat の出力結果の物理メモリ使用量と乖離する原因。
  - 物理メモリの使用量は経験上
    - zfs_arc_max * 2.5 (solaris11.2まで)
    - zfs_arc_max * 5 (solaris11.3以降)

じゃあ、どうするのがいいか？

- nfsサーバ以外のふつうのサーバ
  - ユーザプロセス用に80%のメモリをuser_reserve_hint_pctで割り当てる（というSIerを見たことがある）
  - とすると、カーネル20%なので、これをarcキャッシュとすると（乱暴）、zfs_arc_maxは主記憶実装量の4%以下が適正か？
- nfsサーバ
  - ユーザプロセス用のメモリはほとんど不要で 1-5G位。これはuser_reserve_hint_pctで確保。zfs関連のメモリを全体の50%程度にするなら、zfs_arc_maxに主記憶実装の10%程度の値を設定するか？

そのほか
（zil サイジング）

- 管理情報が書かれる
- 同期ライト(nfs , oracle DBのライト)はライトされるデータそのものも書かれる。
- このディスクがwriteの応答時間を決める。半導体ディスクが良い。
- ５秒に一回データディスクに書き戻される。
- 常に先頭からつかう。（環状には使わない）
- したがって5秒分の書き込みデータが保存できる量で十分。それ以上はムダ
- 経験上、ストライプするとzfsの応答が速くなる。８本までやったことある

そのほか（nfsサーバとして使うときにtip）

- Linuxからnfs　v3 サーバとしてmountするとき
  - noaclオプションが必要
    - cp の-r　オプションでaclを取りに行くのでエラーとなる。
- nfs v4サーバとして使用するとき
  - file 状態が900秒に100万個で固定
  - 沢山ファイルをアクセスすると、次の900秒でファイルstateがガベージコレクトされるまでremote I/O エラーを返す

そのほか
（oracle DBサーバとして使うときのtip）

- zfsではライトが新規セクタに割りつくため、経年変化でディスクI/Oのランダム性が増し、性能低下する。
  - oracle DBを置くファイルシステムとしてufsはがよい。
  - asmは、マルチノード構成でのフェイルオーバ時のデータ保障のため構成が厄介（と聞いた）
- ufsを使うので、キャッシュとして使用されるfreeメモリに十分な量が必要。
- システムディスクはzfsなので、arcキャッシュは必要最低量としてfreeメモリを確保できるようにzfs_arc_maxを設定