NVDIMM向けファイルシステムの開発
AEON: Yet Another File System Designed for NVDIMMs

1　背景

近年，NVDIMMと呼ばれる新しいデバイスの登場によってコンピュータの性能の向上やアーキテクチャ構成の進化が期待されている．NVDIMMは高速かつ不揮発のDIMMモジュールである．これまでの不揮発性の記憶媒体よりも格段に高速でありバイト単位での読み書きが可能であるという点が大きな特徴となる．

NVDIMMは，従来のコンピュータシステムの性能向上に大きく貢献することが期待される一方で，ソフトウェアからの扱いの難しく，従来のソフトウェアではNVDIMMの性能を十分に引き出せていないことが知られている．そのため，NVDIMMを用いたコンピュータの性能向上のための様々な研究や開発が世界中で行われている．

2　目的

本プロジェクトでは，カーネルの機能の一つであり，記憶媒体と最も密接な関係を持つソフトウェアであるファイルシステム「AEON」をNVDIMM向けに開発する．ファイルシステムはLinuxカーネルのファイルシステムとして実装する．NVDIMMは不揮発性であるという特徴からデータを保存するための記憶媒体としての利用が想定されることが多い．ファイルシステムを開発する意義は，個々のソフトウェア資産の特別なチューニング無しにファイルシステムを入れ替えるだけでNVDIMMの恩恵を享受できるということにある．さらに，NVDIMMを記憶媒体として利用する際ファイルを扱う全てのソフトウェアの基盤となるカーネルのファイルシステムがNVDIMMの特性を活かしたものであるかどうかは，より性能の高いコンピュータシステム実現のために重要な要素となる．この基本ソフトウェアであるファイルシステムの開発によってコンピュータ全体の性能の向上を目指し，コンピュータを使用する様々な開発・研究領域のさらなる発展の基礎となることが，本プロジェクトの最大の目的である．

3　開発内容

AEONの全体構成として，先頭ブロックをスーパーブロックと予約inode用に使用し，残りNVDIMM領域をCPUコア数分だけ分割し，各分割した領域に対して管理テーブルを持たせる．分割した領域に対してCPUコア数分フリーリストを持つことにより複数の割当てにスケーラブルに対応できるようにする．NVDIMMはマウント時にカーネルアドレス空間にマッピングされる．AEONはNVDIMM全体の領域を先頭アドレスからのオフセットで管理する．ブロック割当ては実行中のCPUをヒントにフリーリストが選ばれ，そこから割当てられる．

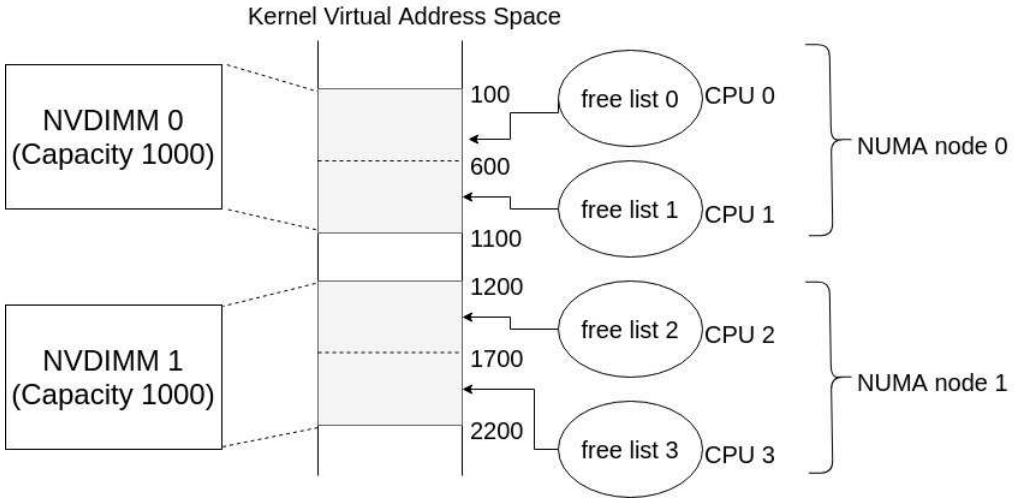

図 1: NVDIMM 領域マッピングと領域管理

図 1 のように Non-Uniformed Memory Access（NUMA）構成のサーバでは，NVDIMM が複数存在する．このような場合は，個々の NVDIMM を別々にカーネルアドレス空間にマッピングする．フリーリストはこちらでも CPU コア数分だけ用意する．NUMA 構成のアーキテクチャの場合は，実行中の CPU に近い NVDIMM を管理するフリーリストが選ばれる．この設計により，実行中の CPU コアから速くアクセスできる方の NVDIMM の領域を割当てて書込みを行うことができる．これらの機構によって，各コンピュータ・アーキテクチャいずれの構成においてもスケーラブルかつ効率的なブロック割当てと使用が可能になっている．

メタデータの割当ては，フリーリストから取得したブロックによって生成されたキャッシュを使って割り当てる．キャッシュのサイズはファイルシステムをコンパイルする時に指定することが可能である．各メタデータは固定サイズになっており，スラブアロケータのような形でメタデータを割り当てる．これによって，メタデータ管理を容易にすると共に割当て速度を高めている．開放された領域はキャッシュの時間的局所性の観点から優先的に再利用する設計になっている．

ファイルデータの読み書きは Direct Access for Files（DAX）と呼ばれる技術を使って行う．これは，NVDIMM に対して最適な読み書きをするための技術である．ファイルの読み書き時に発行される read/write システムコールにおいてはカーネルのページキャッシュをバイパスし NVDIMM へ直接書き込む．また，mmap システムコール時は，NVDIMM 領域を直接ユーザ空間からアクセスさせるようになっている．AEON では，カーネル API を用いた DAX 読み書きとは別に，NVDIMM 領域をカーネルアドレス空間上において管理している設計を活かした独自 DAX 読み書き関数も備えている．DAX によって特に同期処理が多いワークロードの性能が大きく改善される．これは，DAX を使用した時と使用しなかった時で見かけ上の処理が同時に終わった場合であっても DAX の場合はデータが永続化されている可能性が高いためである．また，mmap システムコールによる NVDIMM の書込みの際には読み書き処理時におけるコンテキストスイッチによるオーバーヘッドを避けることができる．

NVDIMM は不揮発性メモリであり従来の HDD や SSD と比べると格段に速い．しかし，現代のコンピュータは CPU とメモリの速度差が大きいことからメモリを直接読み書きするのではなく，CPU キャッシュメモリを用いた読み書きを行う．そのため，NVDIMM に対する読み書きにもキャッシュメモリを活用することがファイルシステムの速度性能を向上させるためには必要不可欠である．しかし，キャッシュメモリは揮発性であり，システムダウンによるファイルシステム破損の危険性がある．従って，速度性能のためにキャッシュメモリを使いつつ，ファイルシステム整合性の保護のための機構が依然必要である．AEON は Consistency Without Ordering（CWO）と呼ばれるメタデータを保護する機構を NVDIMM 向けに再設計したものを備えている．CWO では，図 2 のように，各ディレクトリエントリとファイルに対応するメタデータについて相互にポインタを張る．さらに，ファイルオーナやパーミッション，属性についての重要な情報を相互に持たせる．ファイルシステムの更新が中途半端であった場合，相互ポインタを辿ることができなくなる．これによって，システムダウン後にファイルシステムの更新操作が中途半端になってしまいファイルシステムの整合性が崩れた場合でも，再マウント時に不整合を検知し回復することを可能にして，ファイルシステム操作の原子性を確保している．また，この機能について複数のテストを行っており，検知・回復が可能であることを確認している．

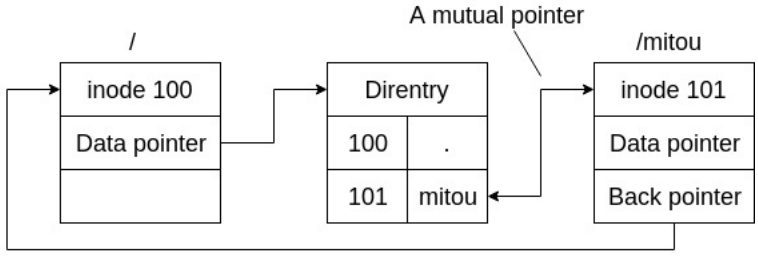

図 2: Consitency Without Ordering for NVDIMMs

AEON は信頼性を保ちつつスケーラブルなシステムにするため，複数ある Linux カーネルのロック API を注意深く使い分けている．処理が少なく，ある程度定まった時間範囲内で終わる処理は spinlocks を用い，状況によって処理の長さが変わるものに対しては mutex locks を用いている．読みと書きで区別をつけれる処理に関しては reader-writer locks を用いて読み込みを並列に行えるようにしている．単一の変数の更新は Atomic operations を用いた．

AEON は速度性能の観点から NVDIMM 領域をカーネルアドレス空間にマッピングしているため，他のカーネルプロセスからの誤った書込みによって領域を汚染されてしまう可能性がある．これを防ぐために Write Protect Control を用いた領域保護を行っている．この領域保護によって AEON が使用する NVDIMM への書込みを禁止している．

AEON は特殊機能として透過圧縮機能を搭載している．透過圧縮アルゴリズムは zstd を利用している．バイトアクセス可能な NVDIMM の特徴を活かすような設計を目指し，オーバーヘッドの少ない透過圧縮を実現している．

4　従来の技術（または機能）との相違

NUMA 構成の NVDIMM-N を 2 基を搭載した HPE ProLiant DL360 Gen10 サーバを用いた性能測定では，マイクロベンチマークで最大 224%，マクロベンチマークで最大 471%，既存のファイルシステムよりも高い性能を示した．比較したファイルシステムは，元々はハードディスク向けであったが NVDIMM 向けのモードでの対応がなされた EXT2，EXT4，XFS，さらに NVDIMM 向けにフルスクラッチされたファイルシステムである NOVA と比較している．いずれのファイルシステムに対しても AEON は速度性能で優れた数値を示した．

AEON とは別の完全な NVDIMM 向けファイルシステムである NOVA との優位点について述べる．AEON は速度性能で NOVA を上回った．さらに，AEON は領域使用量も NOVA と比較して大幅に低く抑えることに成功している．

AEON は，特殊機能として透過圧縮機能を搭載している．これによって，例えば容量の少ない NVDIMM-N について，速度性能低下を抑えつつ領域使用量を減らすことが出来る．

5　期待される効果

NVDIMM はまだ一般的に出回っているものではないが，NVDIMM が一般的なデバイスとして普及してきた時に今回開発したファイルシステムが使用されるようになれることを期待している．その時は，陰ながら様々なコンピュータ・サイエンスの分野の発展の力となれるだろう．

6　普及（または活用）の見通し

本プロジェクトは OSS として GitHub 上に公開されており，現在もバグ修正や機能追加，改善を行っている．今後，ファイルシステムのさらなる信頼性を高めていくことが肝要である．一般的に，意識して使われるソフトウェアというより，気づいたらそこにあるというのが理想である．

7　クリエータ名（所属）

重光 史也（島根大学 大学院総合理工学研究科 総合理工学専攻 情報システム学コース）

（参考）関連 URL

GitHub: `https://github.com/4ge32/aeon`