2004 年度

オープンソースソフトウェア活用基盤整備事業

「OSS 性能・信頼性評価 / 障害解析ツール開発」

ディスク割り当て評価ツール
評価と考察

独立行政法人 情報処理推進機構

商標表記

- Linux は、Linus Torvalds の米国およびその他の国における登録商標あるいは商標です。
- MIRACLE LINUX は、ミラクル・リナックス株式会社が使用許諾を受けている登録商標です。
- その他記載の会社名、製品名は、それぞれの会社の商号、商標もしくは登録商標です。

目次

1. 開発目的とディスク割り当て評価ツールの概要

1.1. 開発目的

サーバ分野において、Linux を適用したシステムが普及・拡大している。数年前までは Web サーバやメールサーバ、ネームサーバといったネットワークのフロントエンドサーバへの適用が中心であったが、最近ではアプリケーションサーバ、DB サーバから構成されるエンタープライズシステムへの適用ニーズも出てきている。

エンタープライズシステムでは迅速な障害対応が求められるが、Linux にはダンプやトレースといった障害解析のための標準的なツールが無く、障害発生時は、各社固有のノウハウで対応しているのが現状である。障害の中でも特に Linux ファイルシステムのディスク割り当てに関しては、フラグメンテーションが起こりにくく障害の原因にはなりにくいといわれており、評価するためのツールもほとんど提供されていない。

しかし最近では、データのアクセス性能が落ちる障害が多々見受けられるようになってきている。この原因のひとつとしてディスクのフラグメンテーションが考えられるが、これを可視化できるツールがないために、アクセス性能劣化障害の原因を切り分けることが困難な状況にあった。

ディスク割り当て評価ツール（Disk Allocation Viewer、以降 DAV と称する）ではこの状況を改善することを目的とした。ディスク割り当て状況を容易に取得／可視化できるようにし、障害原因がフラグメンテーションにある可能性が高いかどうかが分かるようにする。また、ディスク全体のディスク割り当て状況を可視化できるようにしただけでは、アクセス性能劣化が特定ファイルで発生するような場合に原因切り分けが困難になるため、ファイルを対象としたディスク割り当て状況の可視化も可能とする。

1.2. DAV（ディスク割り当て評価ツール）の概要

DAV は、次の 3 つのプログラムおよびカーネルモジュールによって構成される。

(1). ファイルシステム情報取得ツールプログラム

(2). ディスク割り当て可視化ツールプログラム

(3). ファイルブロック情報取得モジュール

ファイルシステム情報取得ツールは、ext2／ext3 パーティションのディスク割り当て状況を取得するプログラムであり、パーティションのマウント状態がマウント／アンマウントいずれであってもディスク割り当て状況を取得できる。また、取得対象としてパーティション全体／単一ファイルを指定できる。

ディスク割り当て可視化ツールは、内部でファイルシステム情報取得ツールを実行し、その結果を GUI によって表示するプログラムである。

ファイルブロック情報取得モジュールは、マウント状態のパーティション上のファイルについてのブロック情報を取得するモジュールである。例えば、ディスク割り当て状況の取得対象ファイルがマウント中のパーティション上に存在する場合は、そのファイルが変更される可能性がある。このような変更がディスク上に書き込まれる以前であっても、ファイル変更後の割り当て状況を取得できるようにするための機能を持っている。

DAV の構成図を**図 1.2-1** に示す。

DAV は対象パーティション／ファイルがアンマウント状態の場合には、ディスク上の i ノード情報を直接読み込み、ディスク割り当て状況を取得する。また、マウント状態の場合にはファイルブロック情報取得モジュール経由でカーネル上の i ノード情報を読み込み、ディスク割り当て状況を取得する。

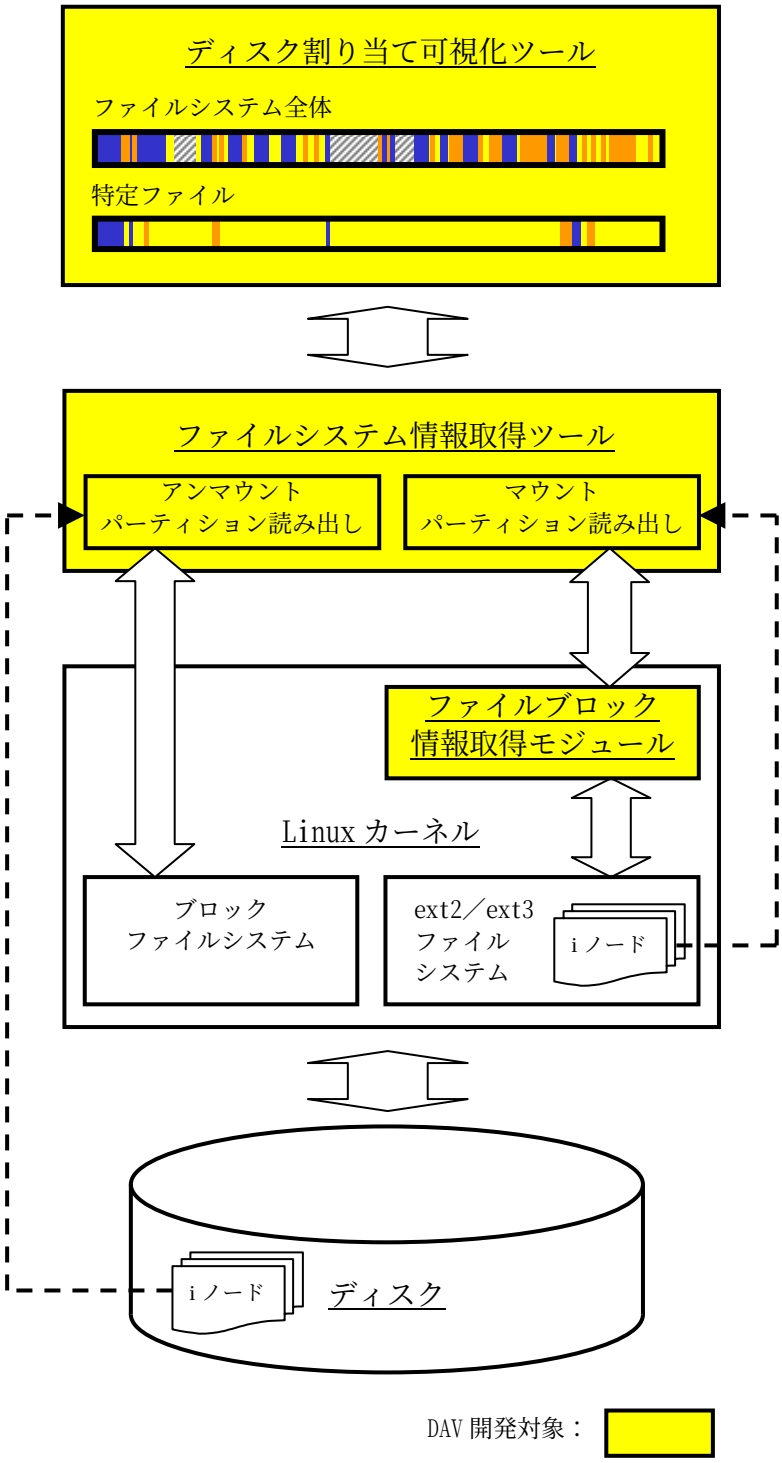

図 1.2-1　DAV 構成

2. DAV によるフラグメンテーション評価

DAV の有効性を確認することと、ディスクフラグメントの影響についてのノウハウを蓄積することを目的として、DAV によるフラグメンテーションの評価を行った。今回の評価項目は以下の通り。

(1). フラグメントによって実際に性能劣化が発生するケースの評価

(2). 既存のデフラグツールを適用した場合のデフラグ効果の評価

2.1. 評価環境

今回の評価環境を**表 2.1-1** に示す。

表 2.1-1　DAV によるフラグメンテーションの評価環境

項番	項目			評価環境
1	ハードウェア	CPU		Pentium４－2.0GHz
2		メモリ		512MByte
3		ハードディスク	モデル	Maxtor 4D060H3
4			容量	60GByte（117187680 セクタ）
5			キャッシュ	2048KByte
6			回転速度	5400rpm
7	ソフトウェア	ディストリビューション		Miracle Linux 3.0
8		カーネル		2.4.21
9		ドライバ		ide-disk v1.17
10		ファイルシステム		ext2／ext3（ブロックサイズ：4KByte）

2.2. フラグメンテーションとアクセス性能劣化の関係評価

ファイル同時書き込みがフラグメント要因のひとつであるということが報告されている。そこでファイル同時書き込みが実際にフラグメント要因となるのか、フラグメントが起こった場合そのフラグメントがどの程度アクセス性能に影響を及ぼすかを評価した。

2.2.1. 評価手順

具体的な評価手順を以下に示す。

(1). ファイルコピー元／コピー先用にそれぞれ 2GByte のパーティションを用意する（同じディスク内のパーティション）。

(2). コピー元ファイルを 32 個用意。各ファイルのサイズは 28～36MByte 間のランダム値で決定。

(3). 親プロセスから fork した子プロセスによってファイル同時書き込み（cp コマンド使用）を行う。同時書き込み数が 2 であれば、fork 子プロセスの数を 2 とし、親プロセスを 32÷2＝16 回分ループすることで 32 個分のファイル全てをコピーする。

(4). 全ファイルコピー終了後のパーティションのフラグメンテーション状況を DAV で取得。

(5). コピー後の各ファイルについてファイル読み出し時間を計測（time cat ファイル名 > /dev/null コマンドを使用）。各ファイルについて、それぞれ 5 回計測する。キャッシュの影響を回避するため、計測のたびにパーティションをマウントし直す。

(6). コピー後の全ファイルを削除。

(7). 上記 (3) ～ (6) の手順を、同時書き込み数を 1、2、4、8、16、32 と変えて繰り返す。

評価手順の概要を**図 2.2-1** に示す。

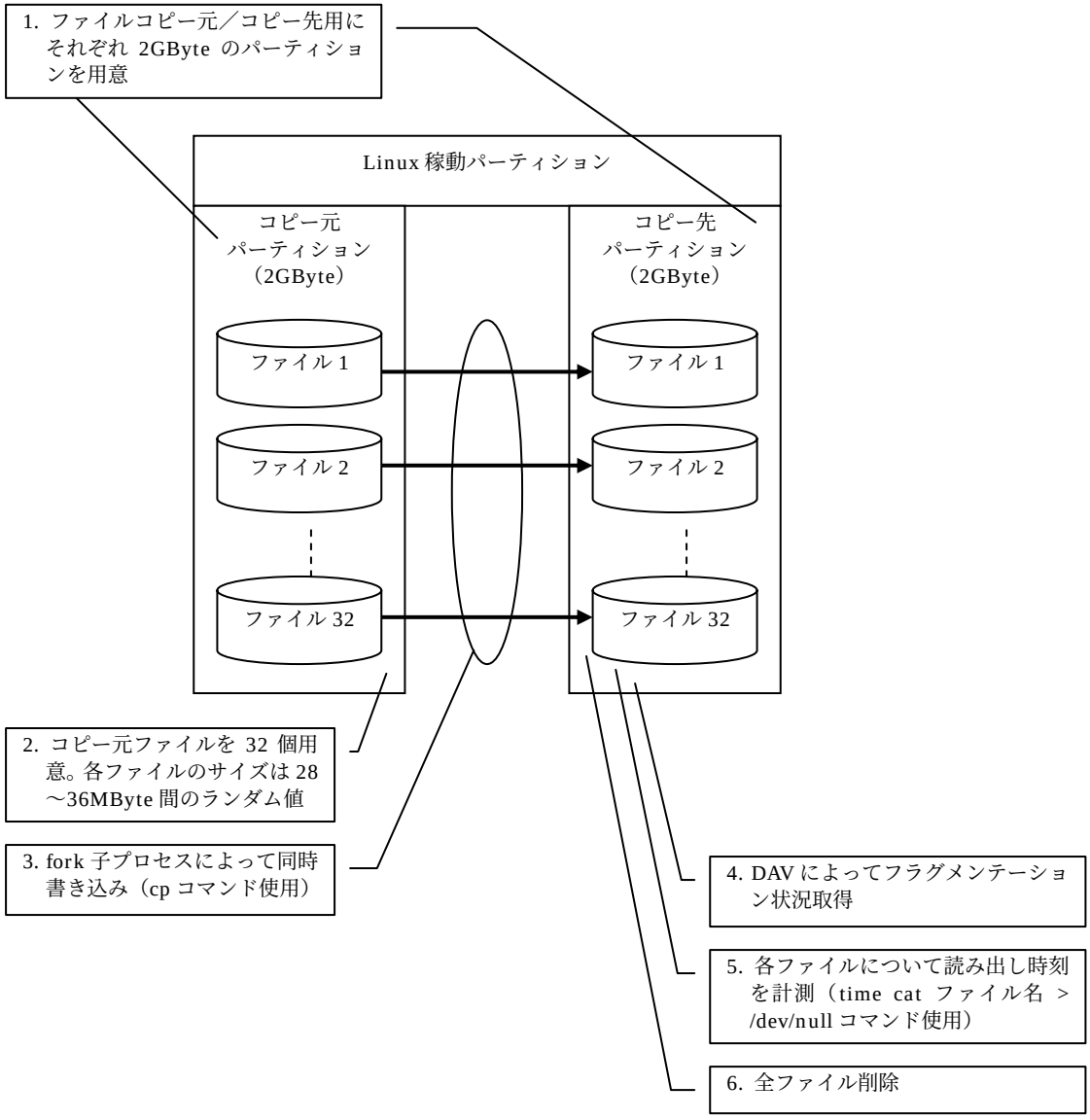

図 2.2-1　ファイル同時書き込みによるフラグメンテーションの評価手順概要

上記手順の(4)で DAV によるフラグメンテーション状況の取得では、今回の開発したファイルを対象としたフラグメンテーション状況取得が効果を発揮した。ディスクを対象としたフラグメンテーション状況取得だけでは各ファイルのフラグメンテーション状況までは取得できないが、この機能によって各ファイルのフラグメンテーション状況を取得できた。

2.2.2. 結果

計測結果の値を**表 2.2-1** に示す。

表 2.2-1　ファイル同時書き込みによるフラグメンテーション数と読み出し時間

項番	同時書き込み数	ext2		ext3	
		フラグ	時間	フラグ	時間
1	1	0.31	1.43	0.31	1.49
2	2	27.06	1.62	34.22	1.68
3	4	51.75	1.78	55.56	1.83
4	8	82.63	2.00	103.84	2.23
5	16	128.72	2.33	138.78	2.46
6	32	153.78	2.42	213.88	2.96

※フラグ：フラグメント数の平均値、時間：読み出し時間（秒）の平均値

ファイル同時書き込み数とフラグメント数の関係を**図 2.2-2** に、フラグメント数とファイル読み出し時間の関係を**図 2.2-3**、**図 2.2-4** に示す。

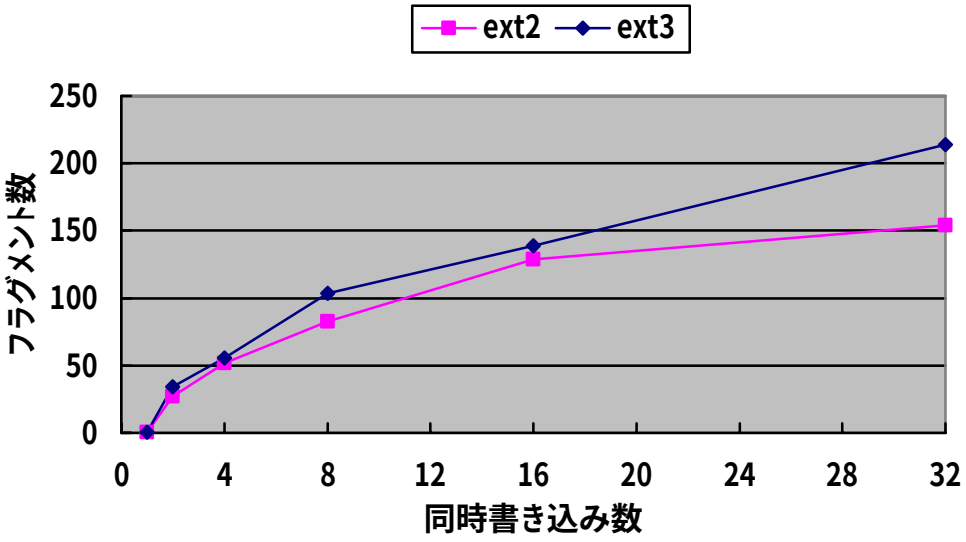

図 2.2-2　ファイル同時書き込み数とフラグメント数の関係

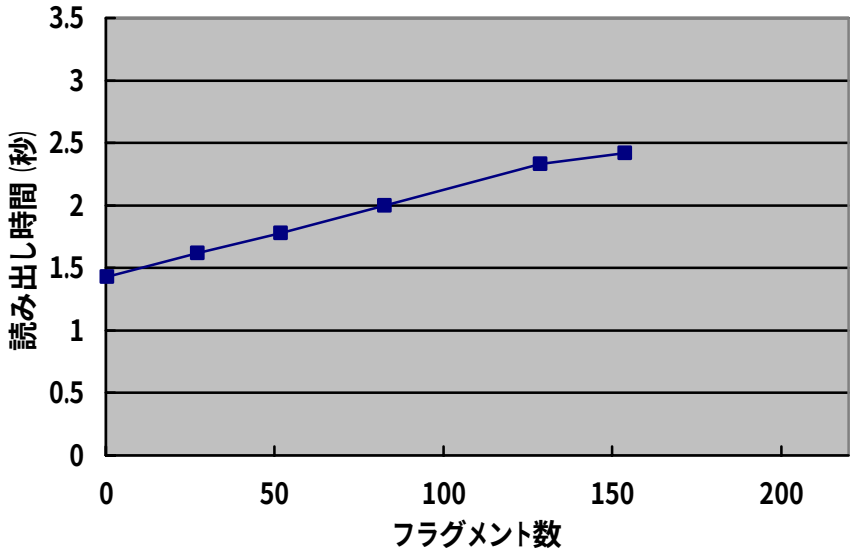

図 2.2-3　フラグメント数とファイル読み出し時間の関係（ext2）

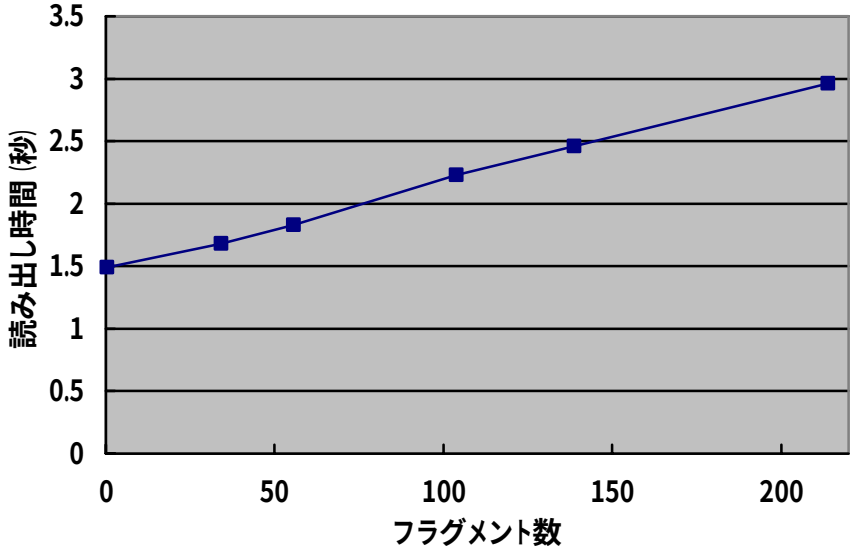

図 2.2-4　フラグメント数とファイル読み出し時間の関係（ext3）

2.2.3. 考察

図 2.2-2 のグラフから、同時書き込み数とファイルのフラグメント数との関係は ext2 と ext3 とでほぼ同じような傾向であり、同時書き込み数が増加するとフラグメント数も増加して行くことが確認できる。また**図 2.2-3**、**図 2.2-4** のグラフから、フラグメント数が増加するとファイル読み出し時間も増加することがわかる。

ext3 の場合、どの程度のフラグメントパーセンテージで読み出し時間がどの程度増加するかを算出する。

まず 1 ファイルあたりのブロック数は下記の算出式によって算出でき、約 8000 ブロックである。

32×1000÷4＝8000（ファイルサイズが平均で 32MByte、ブロックサイズが 4KByte）

ext3 上の 32MByte ファイルの場合、最大のフラグメント数は 213.88 であることからフラグメントパーセンテージは下記の算出式により 2.66%である。

213.88÷8000×100＝2.66%（32MByte ファイルは 8000 ブロック）

フラグメント数が最小と最大の場合で比較するとファイル読み出し時間は下記により約 2 倍となる。

2.96÷1.49　2（最大読み出し時間が 2.96、最小読み出し時間が 1.49）

よって、フラグメントパーセンテージが 2.66%になるとファイル読み出し込み時間は約 2 倍かかっていることがわかる。

これらの評価結果について、Linux では異なるカーネルバージョンでの ext2／ext3 ファイルシステムの変更は激しい（2.6.0～9 とアップデートする間にトータルで 2279step、平均で 253step 変更）ため測定値については評価環境に大きく依存すると考えられる。今回の測定ではフラグメント数に対する読み込み時間については複数回測定し平均値を求めたが、同時書き込み数に関してはそれぞれ 1 回しか測定していないため、複数回計測し平均値を求めると今回の結果とは異なる結果が得られる可能性がある。

これまで Linux ファイルシステム（ext2／ext3）は、Windows と異なりフラグメントが起こりづらいので、定期的なデフラグ実施は不要であるといわれてきたが、上記により使い方次第では簡単にフラグメントが起こり、しかもファイルアクセス性能に大きく影響を及ぼすことがわかる。ファイル容量が大きい場合はファイル読み出しの実時間がかなり長くなると予想され、様々なトラブルの要因に成り得ると思われる。

例えばクライアント／サーバシステムのサーバ側でクライアント要求に応じてファイル書き込みを行うようなケースでは、上記のような同時書き込みが多発すると考えられるため、書き込んだファイルの運用に注意が必要（定期的にバックアップ＆リストアするなど）であると考えられる。

アンマウントパーティションを対象にした場合の DAV 実行時間の参考として、今回計測した際の DAV 実行時間を**表 2.2-2** に示す。

表 2.2-2　DAV 実行時間（アンマウントパーティション）

項番	同時書き込み数	DAV 実行時間（秒）	
		ext2	ext3
1	1	1.72	1.76
2	2	1.99	1.85
3	4	2.05	2.10
4	8	2.24	2.29
5	16	2.38	2.48
6	32	2.85	2.72

※DAV 実行時間は、3 回計測し平均した値

2.3. 既存デフラグツールの効果評価

DAV を用いて、Linux コミュニティに存在するデフラグツールの効果を評価した。

2.3.1. デフラグツールの選定

まず、デフラグ用途で使えそうなツールを調査した。この結果、**表 2.3-1** の２つのツールが候補となった。

表 2.3-1　デフラグ候補ツール

項番	名称	バージョン	ライセンス	URL
1	defrag	0.7x	GPL	ftp://ftp.uk.linux.org/pub/linux/sct/defrag
2	ext2resize	1.1.19	GPL	http://ext2resize.sourceforge.net

defrag は、アンマウント状態の ext2 パーティションをデフラグするためのツールであり、マウント状態のパーティションや ext3 パーティションに関してはサポートしていない。

ext2resize は、ext2／ext3 のパーティションサイズを変更するツールであり、ツール内に含まれるカーネルパッチを適用することでマウントパーティションのサイズ変更も可能にするツールである。パーティションサイズを変更するツールではあるが、パーティションサイズを縮小することでパーティション内に分散したファイルが結合し、フラグメントが解消される可能性があると考えた。

ext2resize を簡単に試したところ、コンパイルとインストールおよび、パーティションの縮小は問題なく実行できたが、パーティション縮小部分以外のファイルについてはそのままフラグメントが残ってしまうことがわかった。本来パーティションサイズ変更ツールであることから、この結果は当然と考えられるため、今回の評価対象からは除外することとした。

defrag は defrag-0.70.tar.gz、defrag-0.73.tar.gz、defrag-0.73-5.src.rpm、defrag-0.73pjm1.tar.gz を試したが、全てコンパイルでエラーとなった。バイナリでは defrag-0.73-5.i386.rpm は依存ライブ

ラリエラーを無視することでインストールできたが、ブロックサイズが 1KByte のパーティションしかサポートしておらず、それ以外の場合にはデフラグが実行されなかった。

最新と思われるバージョン 0.73pjm1 のバイナリパッケージを探したが見つけられなかったため、Debian パッケージのバージョン 0.73pjm1 を流用して評価することとした。

2.3.2. 評価手順

デフラグツールの評価手順を以下に示す。

(1). ファイル同時書き込み数を 32 として、フラグメントを発生させる（具体的な手順は「2.2 フラグメンテーションとアクセス性能劣化の関係評価」を参照）。

(2). デフラグ前のパーティションのフラグメンテーション状況を DAV で取得。

(3). ツールによってデフラグを実行。

(4). デフラグ後のパーティションのフラグメンテーション状況を DAV で取得し、デフラグ前のものと比較。

2.3.3. 結果と考察

測定の結果、デフラグ前後のパーティションのフラグメンテーション状況は、**表 2.3-2** のようになった。

表 2.3-2　デフラグ前後のフラグメンテーション状況比較

項番	項目	ext2		ext3	
		デフラグ前	デフラグ後	デフラグ前	デフラグ後
1	フラグメントパーセンテージ	1.49	0.00	2.06	0.00
2	全ファイルの総ブロック数	328478	328478	331235	331235
3	フラグメント数	4911	0	6834	8
4	システムブロックによるフラグメント数	10	10	10	10
5	所要時間（秒）	233.67		242.96	

※所要時間は 3 回計測し平均した値

システムブロックによるフラグメント数についてはデフラグ前後で変化が見られないが、これは ext2／ext3 の構造上不可避であるため、仕方がない。それ以外のフラグメント数はほとんど 0 になっていることから、デフラグ後はフラグメンテーションがほぼ全て解消されていることがわかる。

図 2.3-1、**図 2.3-2** にデフラグ前後の ext2 パーティションのフラグメンテーション状況を DAV の GUI 画面で表示したものを示す。この GUI 画面からフラグメンテーション状況を可視化することでデフラグ効果が明確になることがわかる。

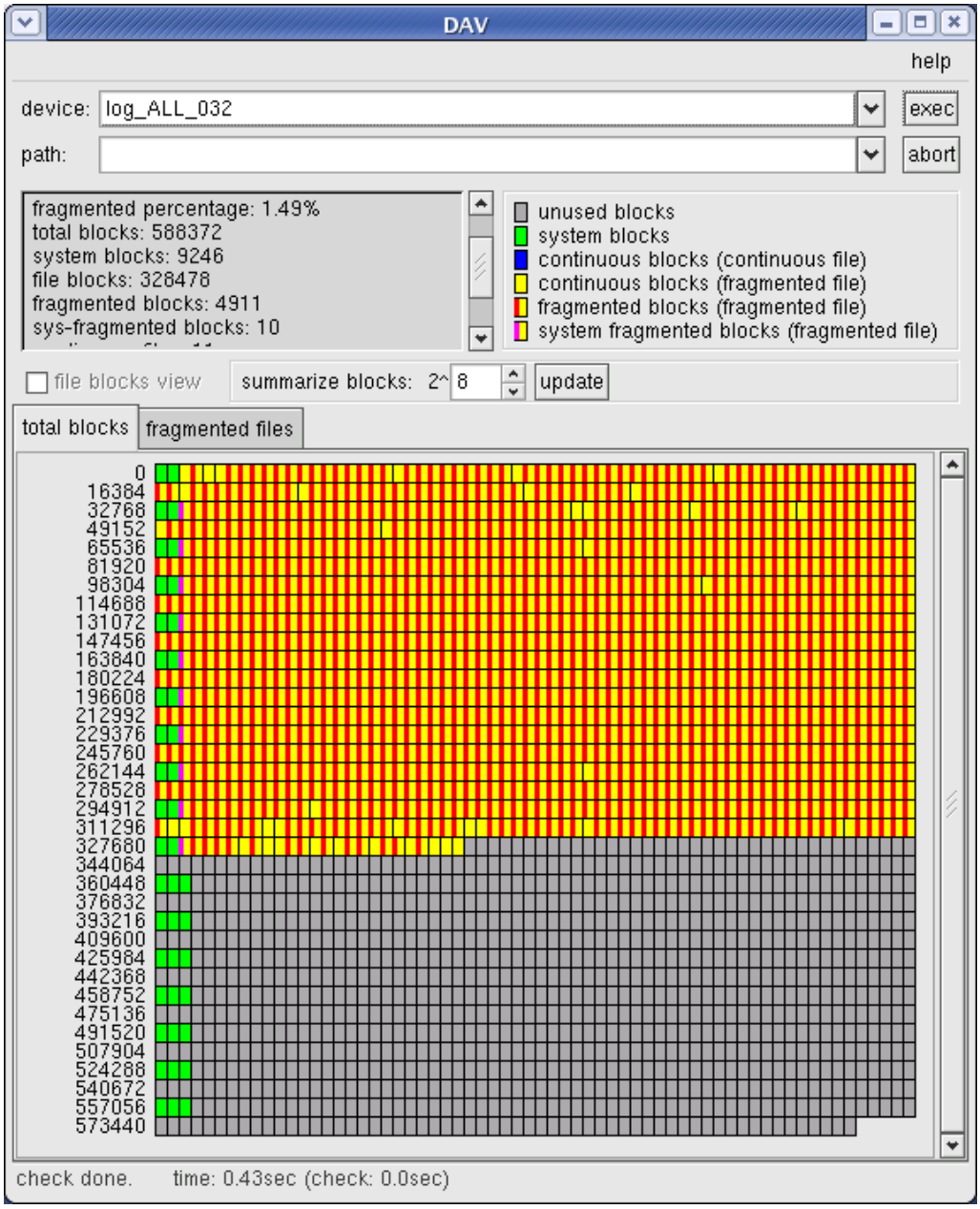

図 2.3-1　DAV によるフラグメンテーション状況表示（デフラグ前）

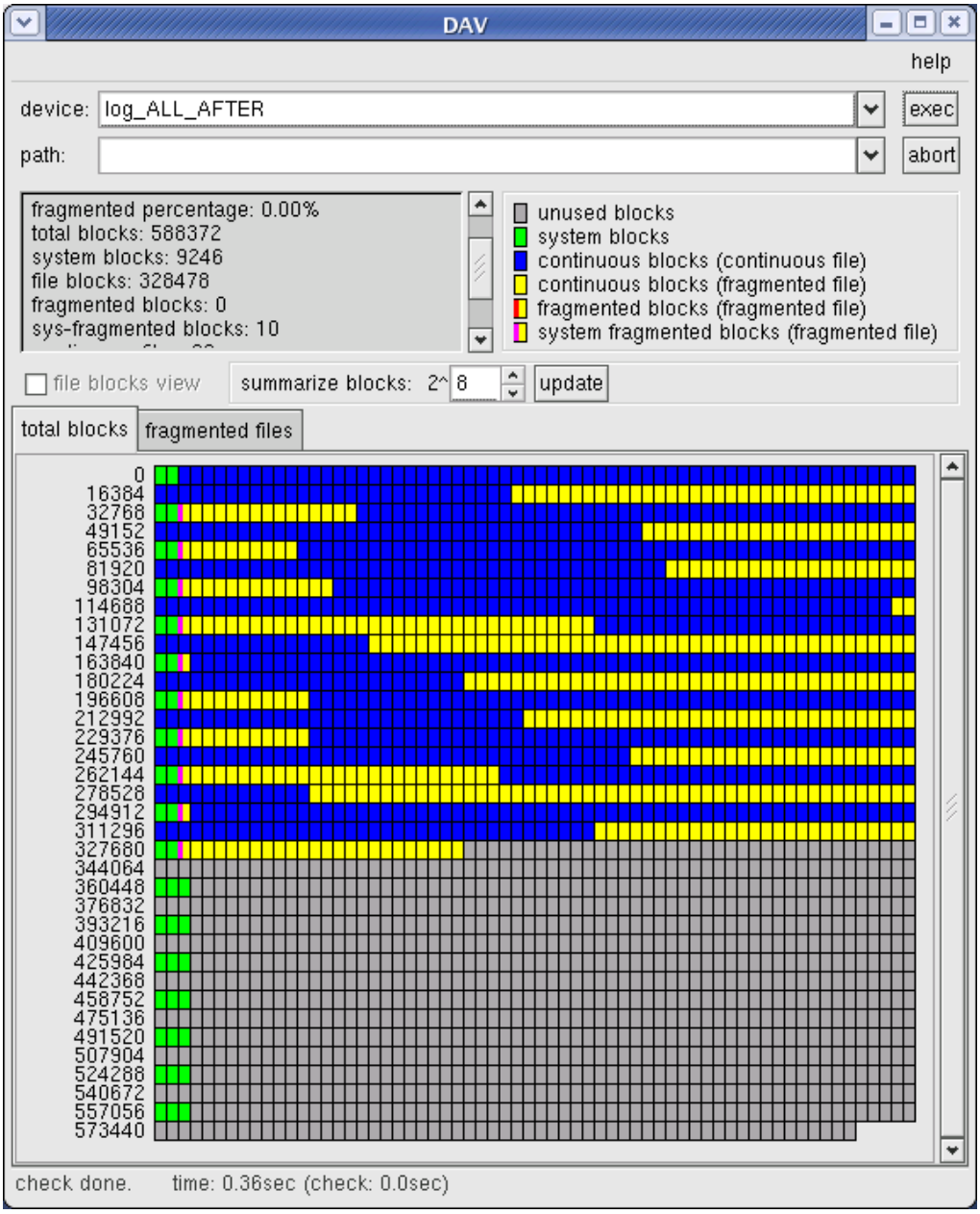

図 2.3-2　DAV によるフラグメンテーション状況表示（デフラグ後）

今回 defrag を評価したが、既に開発が終了しているようで Fedora Core 2 や Miracle Linux 3.0 でコンパイルできるソースは見つけることができなかった。また、評価時は ext3 パーティションに対しても一通りの動作を確認できたが、ext3 サポートは明言されておらず通常運用時の使用には不安が残る。defrag の現在のサポート状況や機能の不足（ext3／マウント状態でのデフラグ実行が未サポート）を考えた場合、新たなデフラグツールが必要である。

3. 総括

3.1. DAV の有効性

今回開発した DAV を利用することで、開発目的であるアクセス性能劣化の原因がディスクのフラグメンテーションにあるかどうかの切り分けを行うことが可能となった。また、ファイル単位でフラグメント数を取得可能であるため、特定のファイルにフラグメントが集中している場合でも問題の切り分けが可能となり、開発目的を満足できたと考える。

これまでの DAV の開発および DAV を用いた評価を通じて、わかった DAV のメリットは次の通り。

(1). 性能劣化や異常終了などが発生した場合、フラグメント発生によるアクセス性能劣化の可能性を切り分けることができる。

(2). ベンチマークなどと組み合わせることで、システムの性能劣化の予想が立てられ、その対応を考えることができる。例えば今回の評価のように同時書き込みによってどの程度フラグメントが発生するかを確認することで、他のカーネルバージョンを使用したり、運用方法を変えたりするなどの対応が可能になる。

(3). フラグメンテーション状況を GUI ブロック表示できるため、フラグメンテーション状況全体が把握しやすく分かりやすい。更にブロックを集約して見たりファイル構成順に並び替えたりすることができるので、多角的に見ることができる。

(4). フラグメンテーション状況はテキスト形式で出力することもできるため、容易に他のコマンドと組み合わせることが可能である。例えば、cron と組み合わせることで定期的にディスクのフラグメント状況を DAV で取得し、性能劣化が発生しつつあることを確認する、などといった応用が可能。

3.2. 開発規模

DAV の開発規模を**表 3.2-1** に示す。

表 3.2-1　DAV 開発規模

項番	プログラム	開発言語	ステップ数
1	ファイルシステム情報取得ツール	C	2,112
2	ディスク割り当て可視化ツール	C	2,823
3	ファイルブロック情報取得モジュール	C	432

3.3. 今後の課題

DAV の開発によって当初の目的は達成できたが、DAV で取得できるのはあくまでも i ノードから見た論理的なブロック位置でのフラグメント数であり、物理的なシーク／アクセス時間とは必ずしも一致しないため、問題切り分け後に詳細な原因特定を行うためには LKST（Linux Kernel State Tracer の略。SourceForge（http://lkst.sourceforge.net, http://lkst.sourceforge.jp）にて公開している）と組み合わせて解析を行うことが理想である。今回の評価ではそこまで実施できなかったため、

今後の課題としたい。

また今回開発した DAV は、次のような改善によって更に活用範囲が広がる。

(1). XFS／JFS／ReiserFS への対応。これを実現することにより、用途に応じてファイルシステムを選択する場合などに候補が増えるため、検討の幅が広がる。

(2). どのディレクトリにフラグメントが集中しているかといった情報の可視化。これを実現することにより、パーティション内でフラグメントがある箇所に集中している場合に、その箇所を特定する作業がさらに高速化できる。

(3). 利便性の向上。例えばパーティション一覧の表示や、フラグメントファイル一覧での各種条件によるソートなどを実現することで、使い勝手が向上しフラグメント状況を多角的に見られるようになる。

これら改善を行うことで、対象となるシステム／ユーザを拡大して行くことが今後の課題である。

本書は、独立行政法人 情報処理推進機構から以下の８社への委託開発の成果として作成されたものです。
委託先企業：（株）日立製作所（幹事会社）
（株）SRA、（株）NTT データ、新日鉄ソリューションズ（株）、
住商情報システム（株）、（株）野村総合研究所、ミラクル・リナックス（株）、
ユニアデックス（株）　（五十音順）