フラグメンテーション解析を支援するDAVとLKST Log Toolsの開発

平松 雅巳†, 杉田 由美子†, 藤原 哲‡

†(株)日立製作所　システム開発研究所,
‡日立ハイブリッドネットワーク株式会社

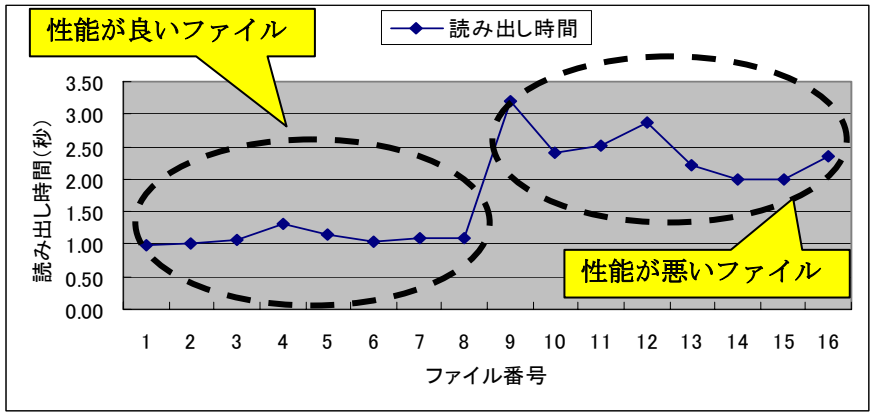
目次

- はじめに
  - 背景
  - フラグメンテーションの問題
- ツールの開発
  - DAV
  - LKST Log Tools
- フラグメンテーション影響の解析
  - フラグメンテーションの作成
  - DAVによる解析
  - LKST Log Toolsによる解析
  - 2つのツールのデータの整合性評価
- まとめ

- はじめに
  - 背景
  - フラグメンテーションの問題
- ツールの開発
- フラグメンテーション影響の解析
- まとめ

背景

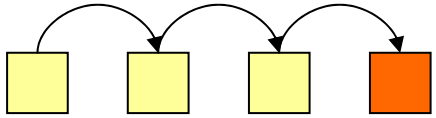
- 事例1：巨大ファイルの上書きバックアップ
  ➡運用していると，次第に時間がかかるようになる
- 事例2：並列書き込みしたデータの読み出し
  ➡ファイルによって読み出し性能にばらつきがある



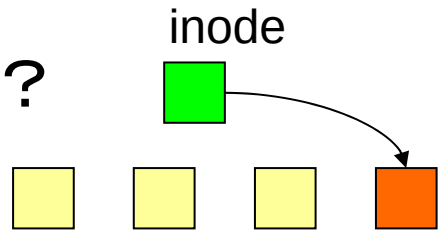
フラグメンテーション（ファイル断片化）の問題?

フラグメンテーションの問題（1）

- DOS（FAT）では昔から問題
  - チェーン構造によるファイルブロック管理が原因
    - 部分アクセスで，チェーンを辿る必要



- Linuxでは？
  - inode管理⇒Ext2/Ext3では問題ない？

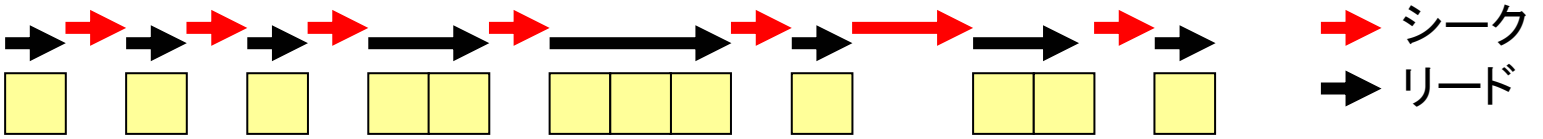


  - フラグメンテーション情報収集・表示ツールが無く、問題がないのかを確認できない
    - fsck：断片化したファイルの数/ファイル総数のみ表示

フラグメンテーションの問題(2)
― 性能への影響 ―

- 連続したファイル
  - データが一箇所に集中
  - アクセス効率が良い(特にシーケンシャルリード)

- フラグメンテーションが発生したファイル
  - データが細分化され、広域に分散
  - アクセス性能が悪くなる(特にシーケンシャルリード)



フラグメンテーションの解析

- 2つの視点からの解析
  - 断片化状態の解析
  - 性能への影響解析

→それぞれの解析ツールを開発

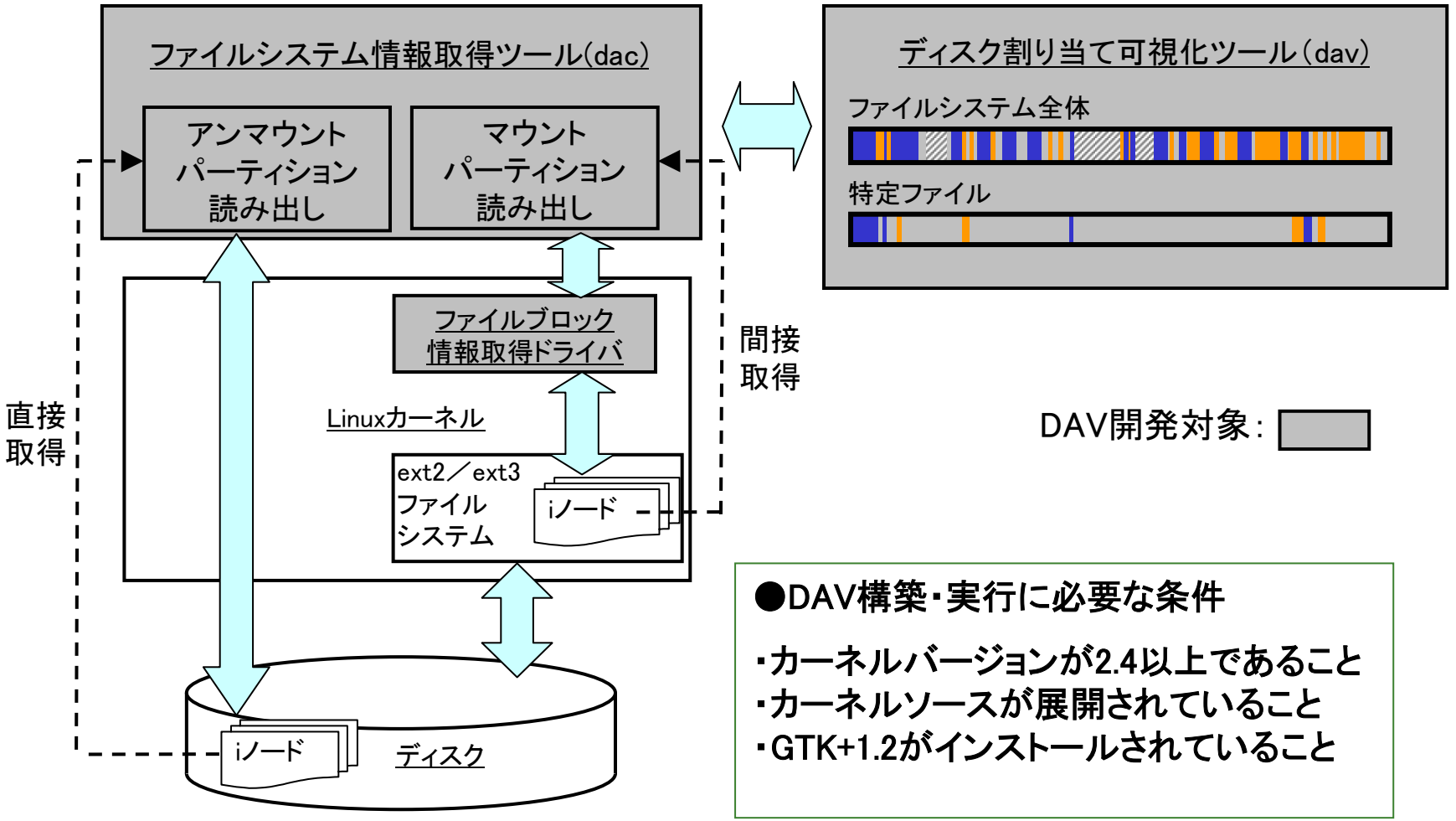
- 静的解析ツール　DAV
  - ファイルの断片化状態の情報を収集し、表示する
- 動的解析ツール　LKST Log Tools
  - カーネル内のI/O処理の情報を収集し、解析・表示する

- はじめに
- ツールの開発
  - DAV
  - LKST Log Tools
- フラグメンテーション影響の解析
- まとめ

静的解析ツール　DAV（Disk Allocation Viewer）

- 目的
  - ディスク上に配置されたファイルの断片化状態の表示
- 仕様
  - Ext2/Ext3に対応
  - マウント・アンマウントにかかわらず状態を解析
    - マウント中の解析にはドライバが必要
  - パーティション・ディレクトリ・ファイル単位で解析可能
- プログラム構成
  - dac: CUI（テキスト出力）の解析ツール
  - dav: GUIの解析結果表示ツール
  - dav-liveinfo：解析補助ドライバ

DAVの構成



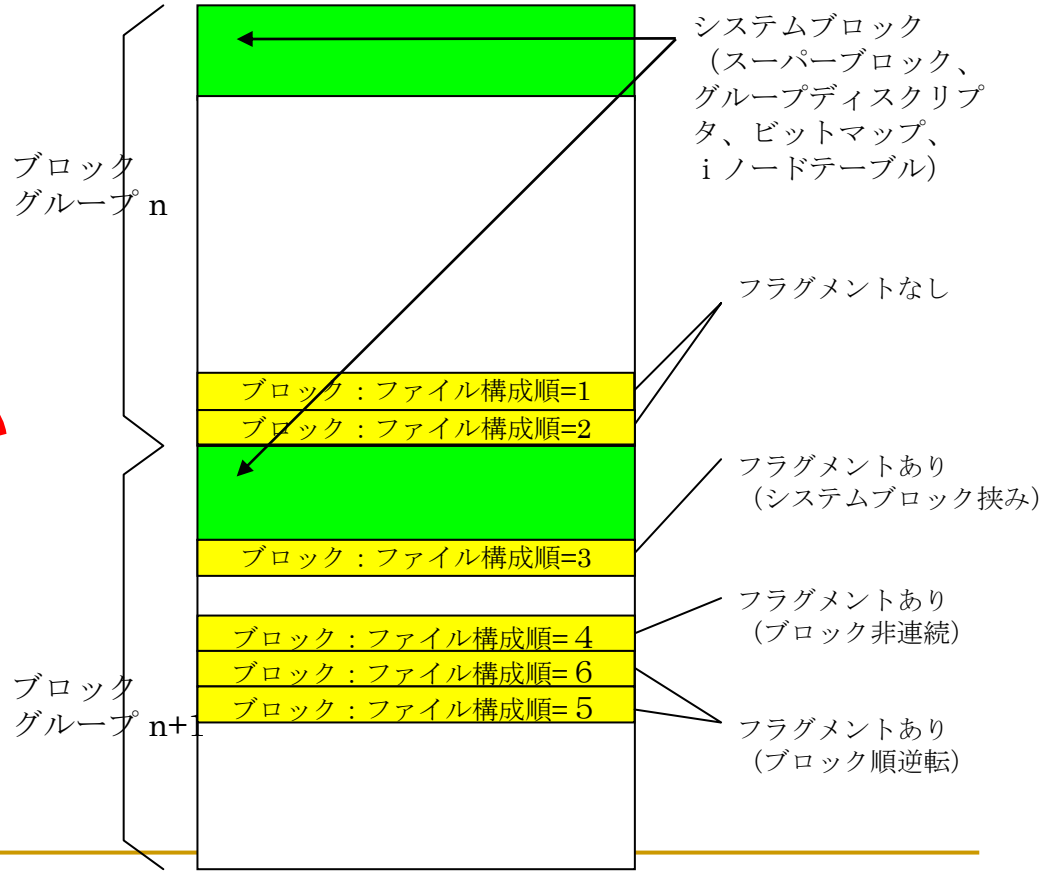
●DAV構築・実行に必要な条件

・カーネルバージョンが2.4以上であること
・カーネルソースが展開されていること
・GTK+1.2がインストールされていること

フラグメンテーションの判定仕様

ファイルブロックをファイル構成順に見て行き、下記の状態の場合に、フラグメントブロックと判断。

(1). ファイルブロックが、
システムブロックを挟んでいる
(2). ファイルブロックが、
物理的に連続していない
(3). ファイルブロックの
順番が逆転している

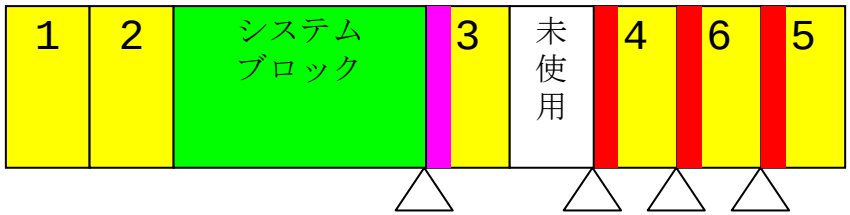


フラグメンテーションの表示仕様

フラグメンテーション状況の出力（テキスト形式／GUI表示）は2種類

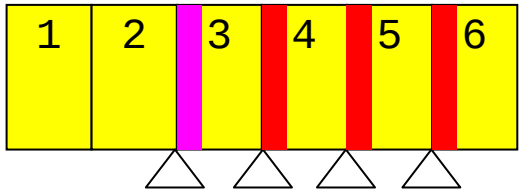
(1).ブロック番号順表示

物理的なブロック番号順に，システムブロックや未使用ブロックを含めた表示



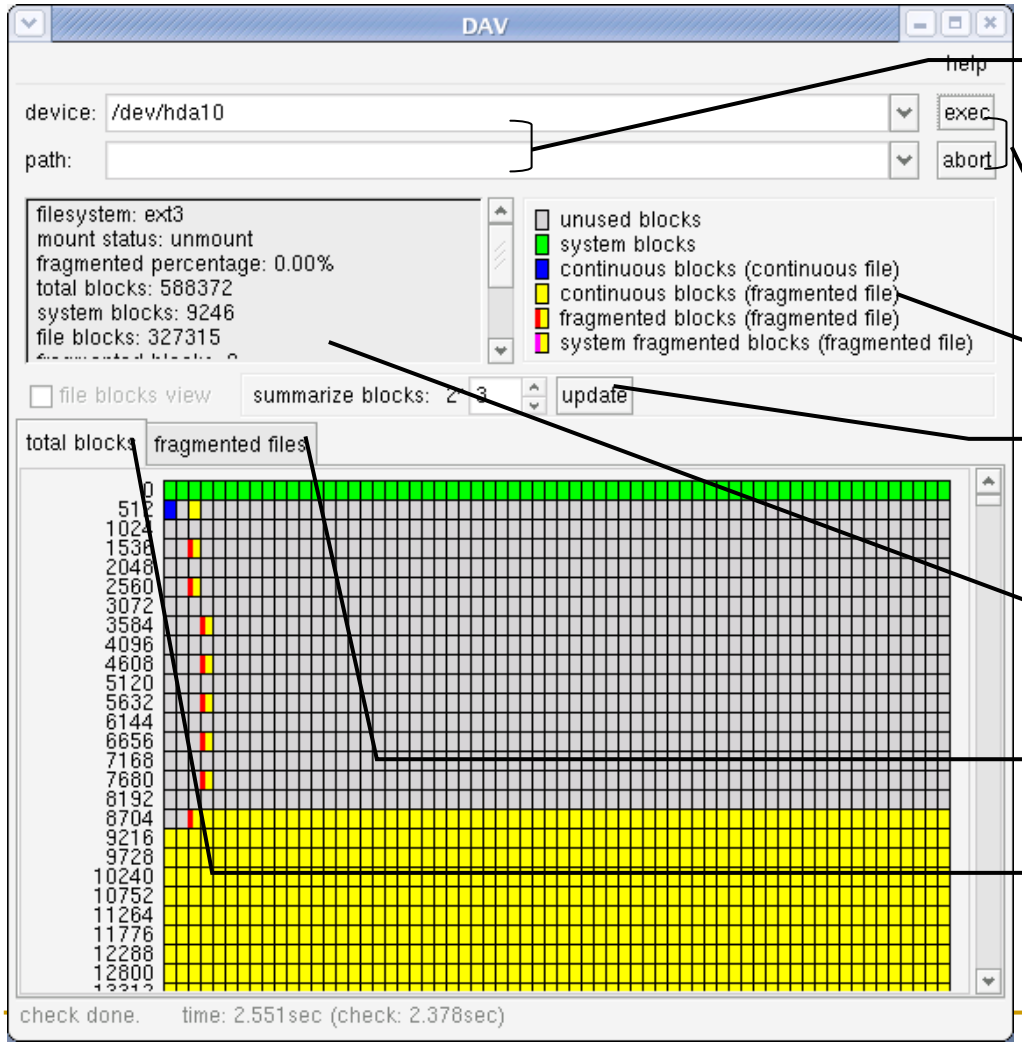
(2).ファイル構成順表示

確認対象のファイルを構成するブロックだけを、ファイル構成順に表示



※数字は、ファイル構成順番
△は、フラグメント位置

DAVの出力形式（GUI）



フラグメンテーション状況確認対象の指定

フラグメンテーション状況取得の開始／中止を指定

ブロック表示の色分けの凡例

何ブロック分を集約して1つのブロックに表示するかを指定

フラグメンテーション状況のテキスト出力部分

フラグメントファイル表示タブ

ブロックデバイス全体表示タブ

- はじめに
- ツールの開発
  - DAV
  - LKST Log Tools
- フラグメンテーション影響の解析
- まとめ

動的解析ツール　LKST Log Tools

- 目的
  - カーネル内の性能情報を収集・解析し，評価を支援
- 仕様
  - 性能評価対象ごとに個別の性能指標値を算出
  - データの切り出し・表示形式の変更が可能
    - フィルタ・フォーマッタの切り替え
  - 解析結果をプロット
- プログラム構成
  - LKSTの機能拡張ドライバ＋パッチ
  - 性能データの解析ツール
  - 解析結果のプロットツール

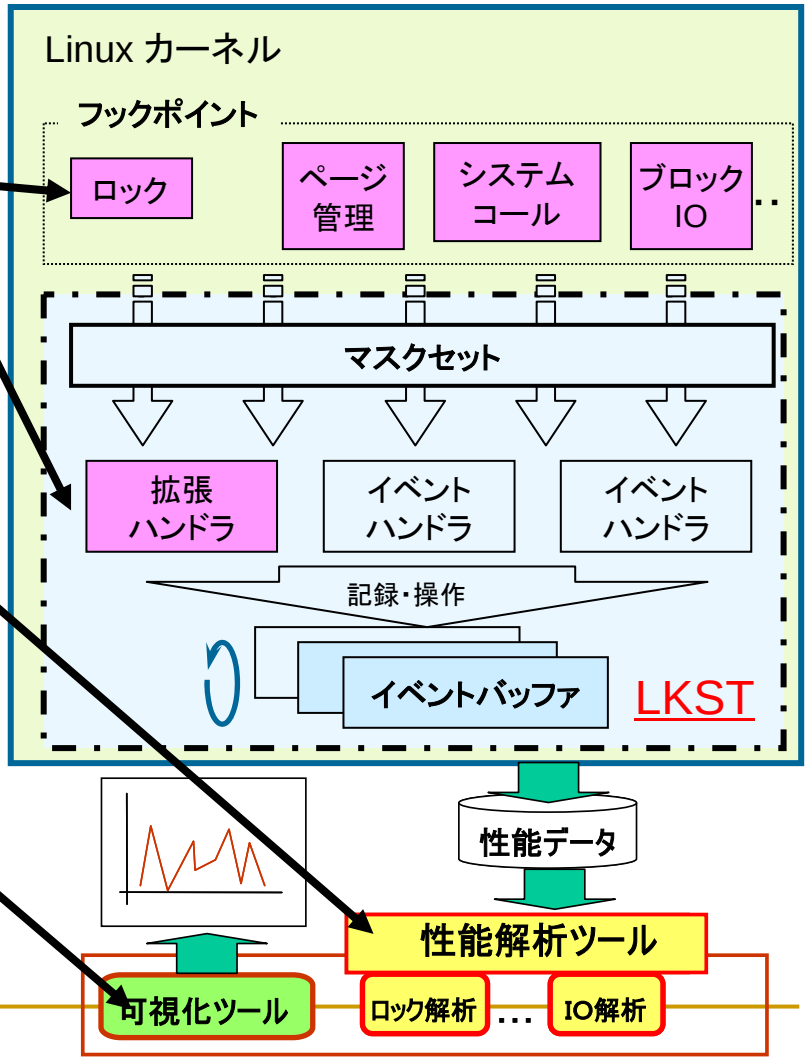
LKST: Linux Kernel State Tracer

LKST Log Tools（カーネル性能評価機能）

■性能評価ポイントで情報を取得する
「LKST機能拡張」ドライバ＋パッチ

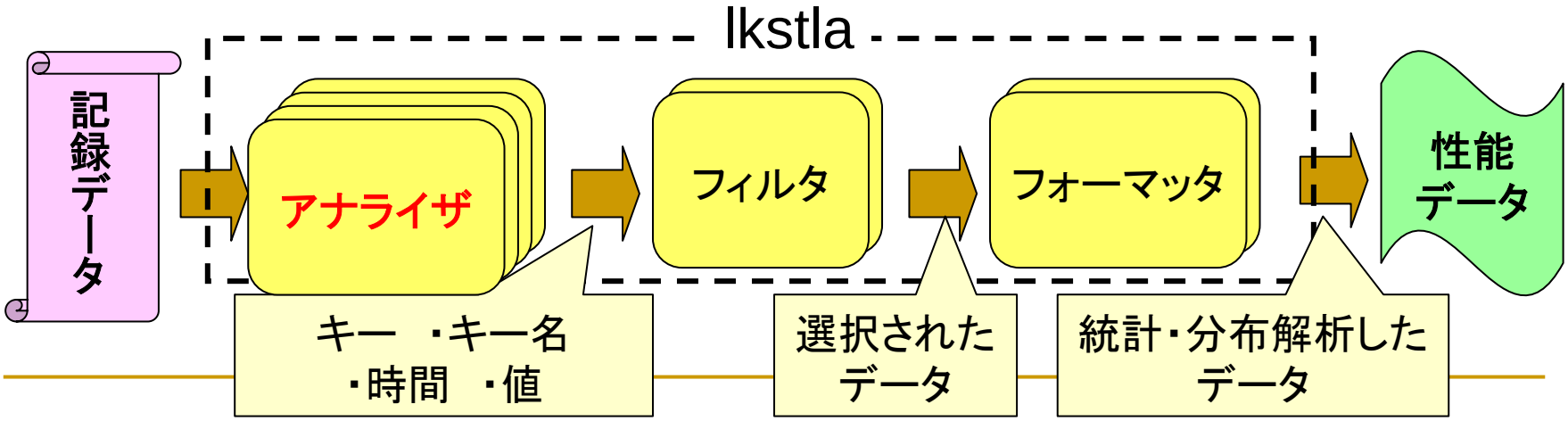
■取得した情報を解析する
「性能解析ツール:lkstla」

■解析した情報を可視化する
「可視化ツール」



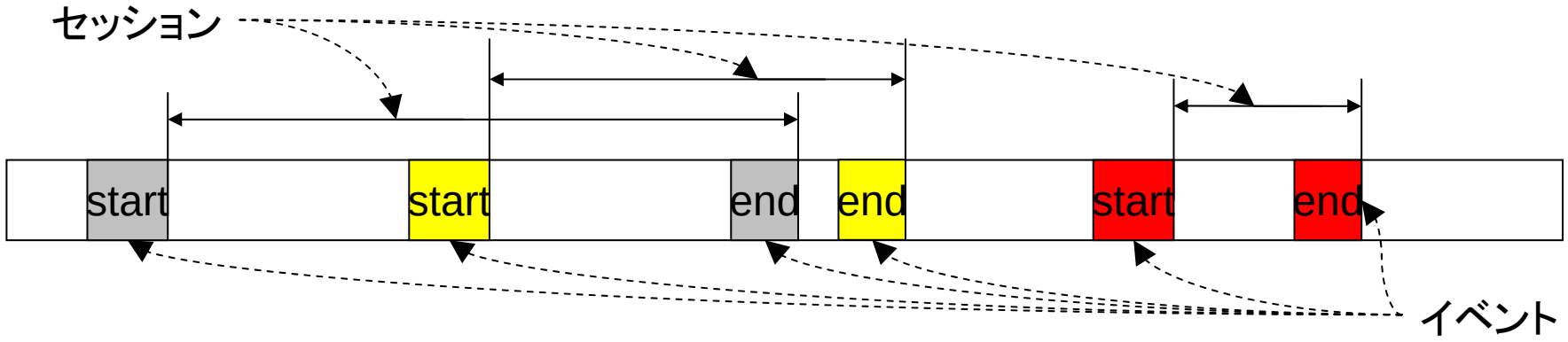
性能解析ツール(lkstla:lkst log analyzer)

- 3つの機能から構成
  - アナライザ(28種類)
    - LKSTのログファイルを解析して、性能を示す値を抽出する。
  - フィルタ(5種類)
    - いくつかの条件によって、結果を選り分ける。
  - フォーマッタ(3種類)
    - 解析結果の統計や分布などを表示する。



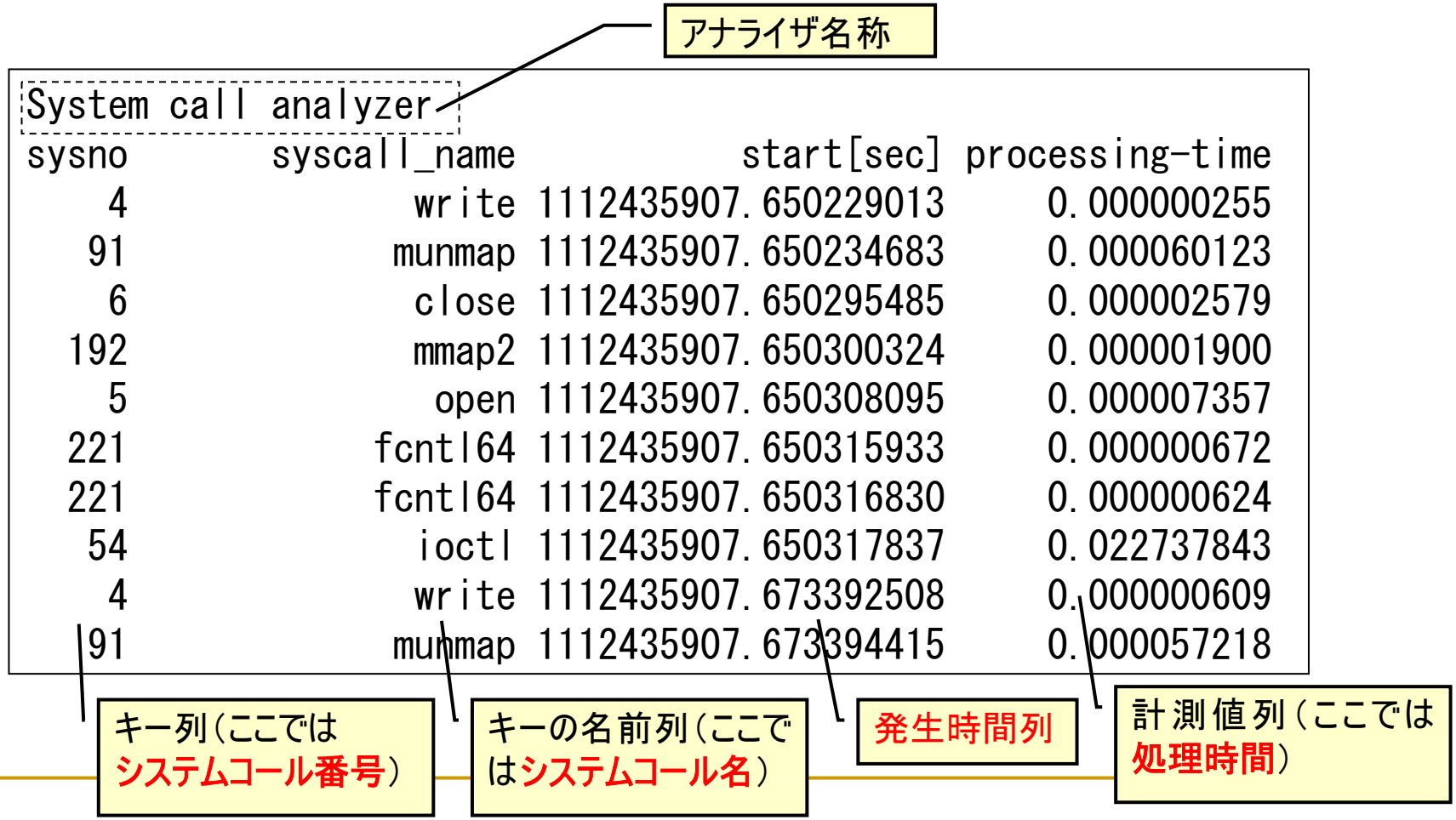
性能解析ツール(lkstla)のアナライザ

- LKSTで記録したログファイルを解析
- 二種類のアナライザに大別
  - セッションアナライザ
    - 二つ以上の関連するイベントが発生した時間の間隔を測定
    - 例）システムコールの入口・出口
  - イベントアナライザ
    - あるイベントの引数に注目し，解析する
    - 例）プロセスの状態遷移



lkstlaの出力形式

- 時系列フォーマットでの出力例(syscall)

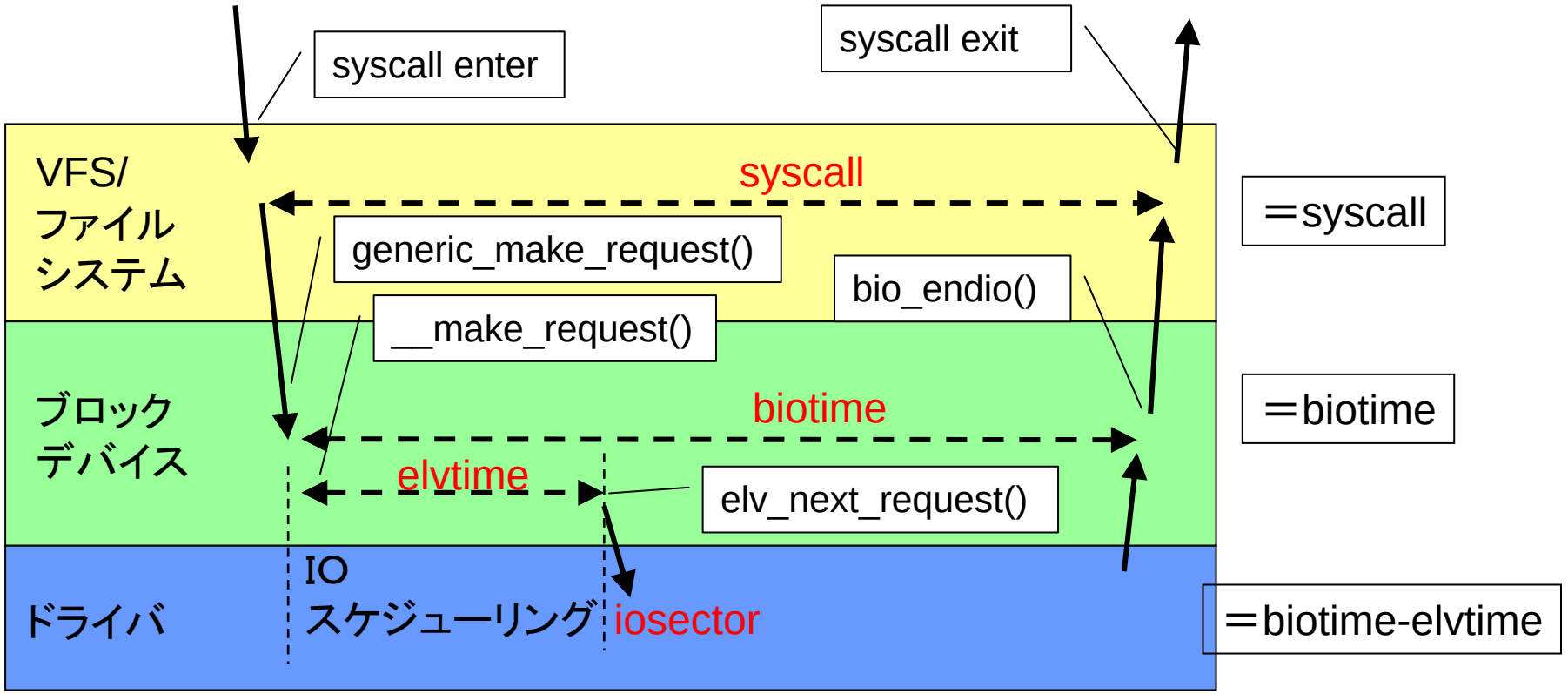


LKST Log Toolsによるファイル IO解析とは(1)

- ファイルI/Oの処理層ごとの処理時間を計測
  - syscallアナライザ
    - システムコールの処理時間
    - Wait状態も含む. (ディスクシーク時間などの影響も加味)
  - biotimeアナライザ
    - ブロックデバイス層に渡されたBIOの処理時間
    - BIOごとに算出
  - elvtimeアナライザ
    - IOスケジューラにリクエストが滞留している時間
    - IOリクエストごとに算出
  - iosectorアナライザ
    - デバイスドライバへ送られるIOリクエストのセクタ番号
    - IOリクエストごとに算出

LKST Log Toolsによるファイル IO解析とは(2)

- ファイルI/O処理層



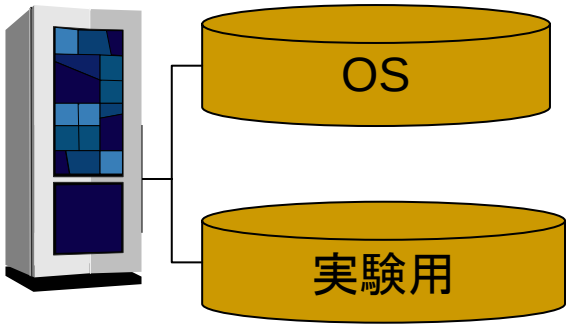
- はじめに
- ツールの開発
- フラグメンテーションによる影響の解析
  - フラグメンテーションの作成
  - DAVによる解析
  - LKST Log Toolsによる解析
  - ２つのツールのデータの整合性評価
- まとめ

フラグメンテーションによる影響の解析

- 目的
  - フラグメンテーションの状況・影響の解析
- 実験・解析手順
  1. フラグメンテーションの作成
  2. DAVによる静的解析
  3. フラグメンテーションファイルのベンチマーク
  4. LKST Log Toolsによる動的解析
  5. 2つのツールのデータの関係

実験環境

- 実験用PC
  - Dual Xeon 3.6GHz（HT有効）
  - Memory 8GB
  - Ultra 160 SCSI Adaptor
  - U 320 SCSI Disk 10025rpm x2（OS用・実験用）



- ソフトウェア
  - Fedora Core 2
  - Linux 2.6.9 + LKSTv2.3
  - DAV v1.0.1
  - IOzone benchmark ver 3.233

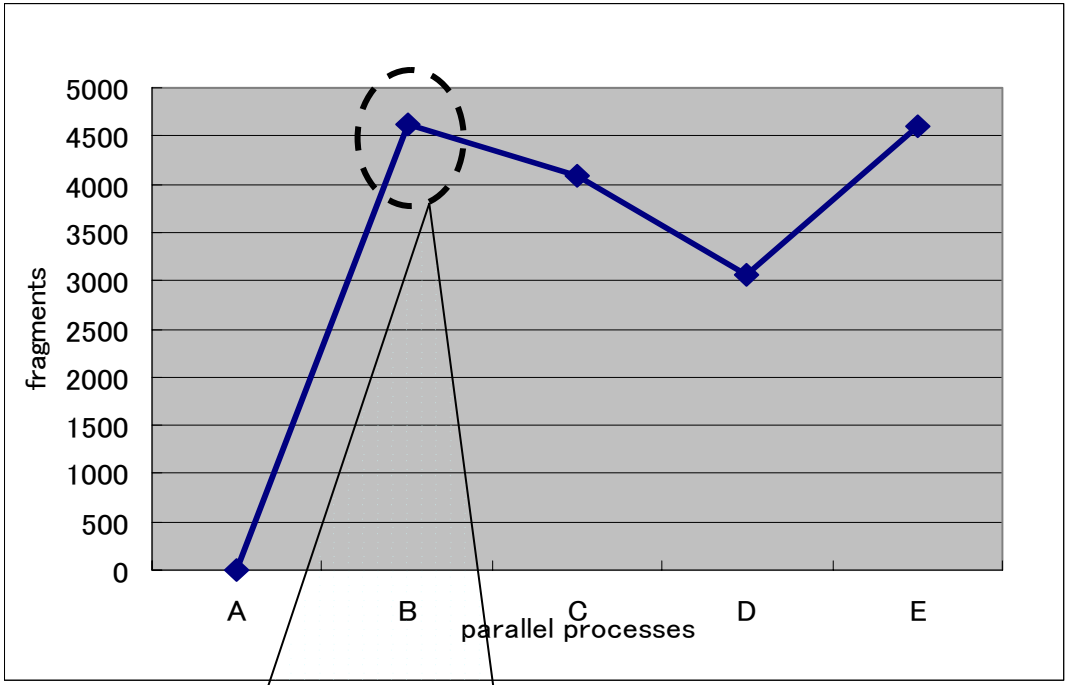
フラグメンテーションの作成

- IOzoneを使い，書き込みの並列度を上げる
  - IOzone パラメータ

パターン	ファイル サイズ	書き込み サイズ	並列 プロセス	並列作成 ファイル数	作成 回数
A	20MB	1MB	1	1	16
B			2	2	8
C			4	4	4
D			8	8	2
E			16	16	1

DAVによる解析（1）– 並列度と断片数の関係 –

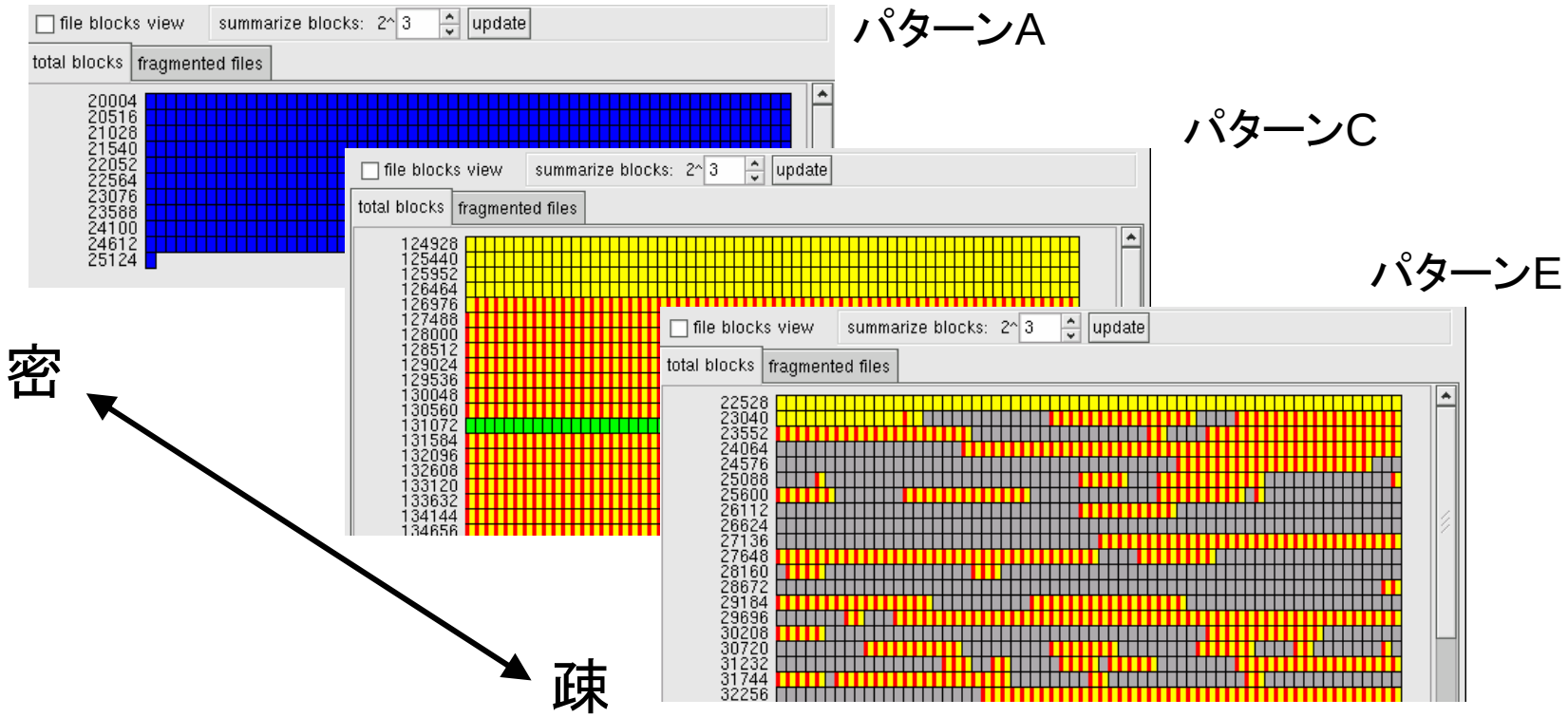
- dacで取得したパターンA~Eのフラグメント数
  - フラグメント数の平均を算出



２並列から既に断片化している

DAVによる解析（1）－断片化状態－

- 各パターンの解析結果(代表的ファイルの解析)



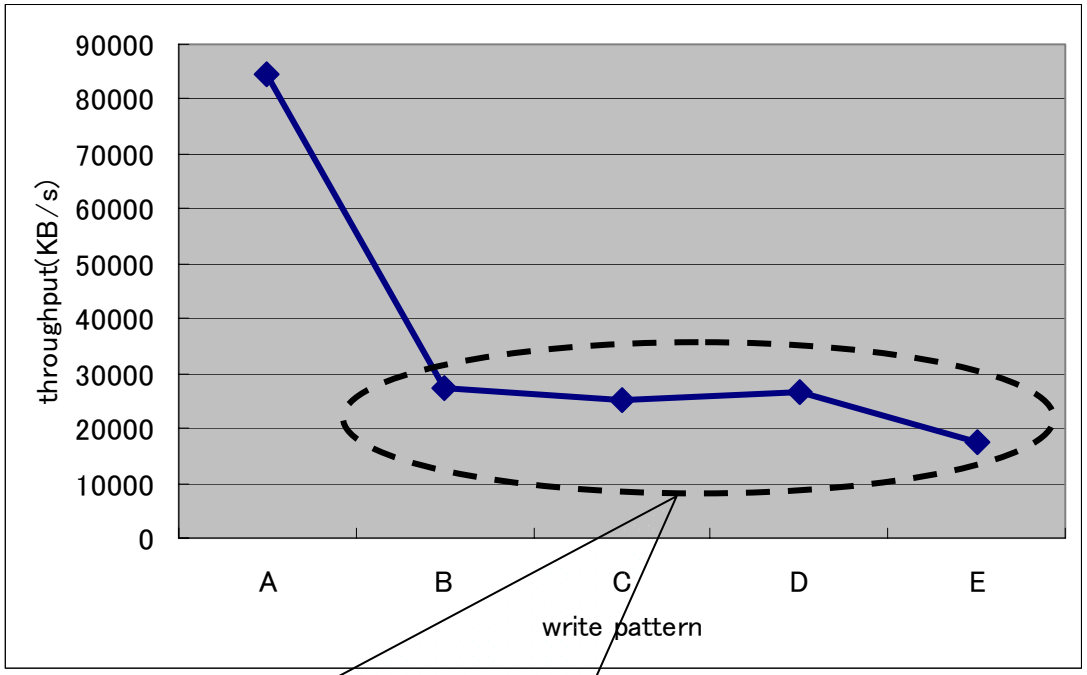
→並列度が上がると，フラグメントが発生し，ブロックが分散する

フラグメンテーションの性能測定

- IOzoneを使い，ファイルを読み出す
  - テスト毎に対象デバイスを再マウント
    - 巨大ファイルのバックアップを想定
  - 各パターンのファイルをシーケンシャルリード
    - フラグメンテーションの影響を最も受けると予想
  - 読み出しの並列度は1プロセス
    - スケジューラの影響を抑える
  - 1MB単位のreadを20回発行
- LKSTによる測定
  - IO処理各部への影響を解析

フラグメンテーションの性能測定結果

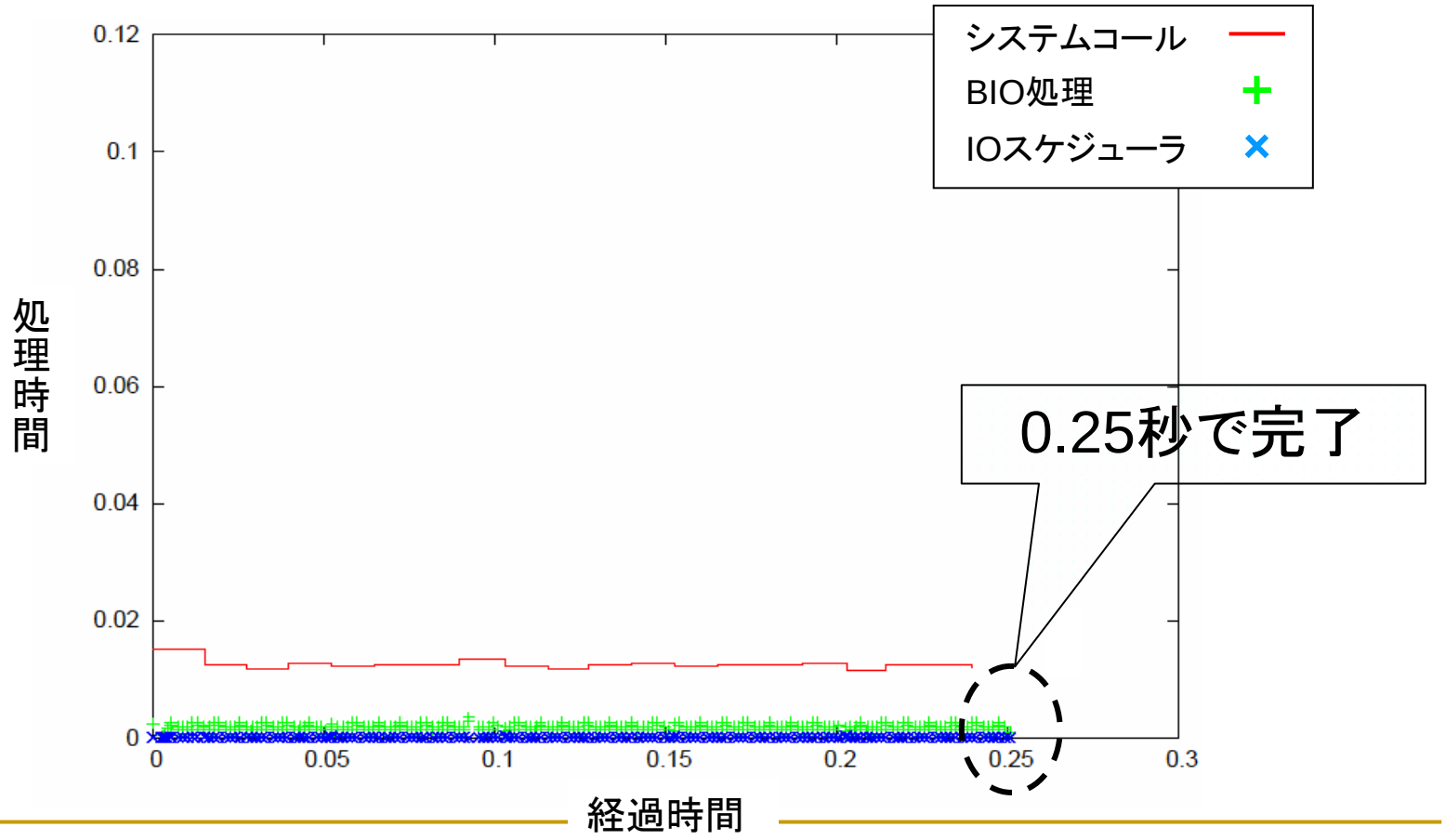
- パターンA~Eのスループット
  - スループットの平均を算出



断片化しているだけでスループットが低下

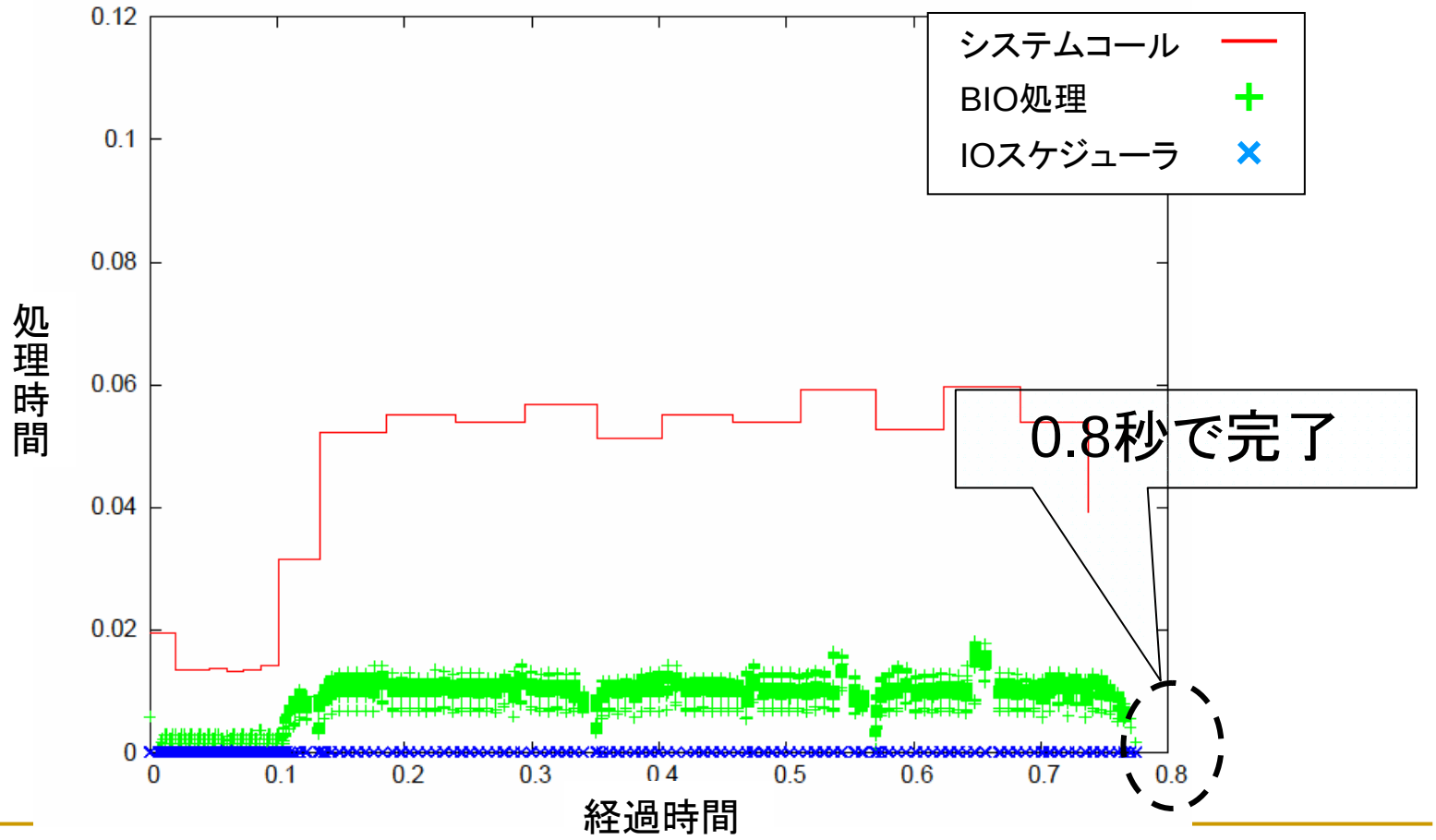
LKSTによる解析(1-1) - パターンA -

- 全てのレイヤで一定の処理時間



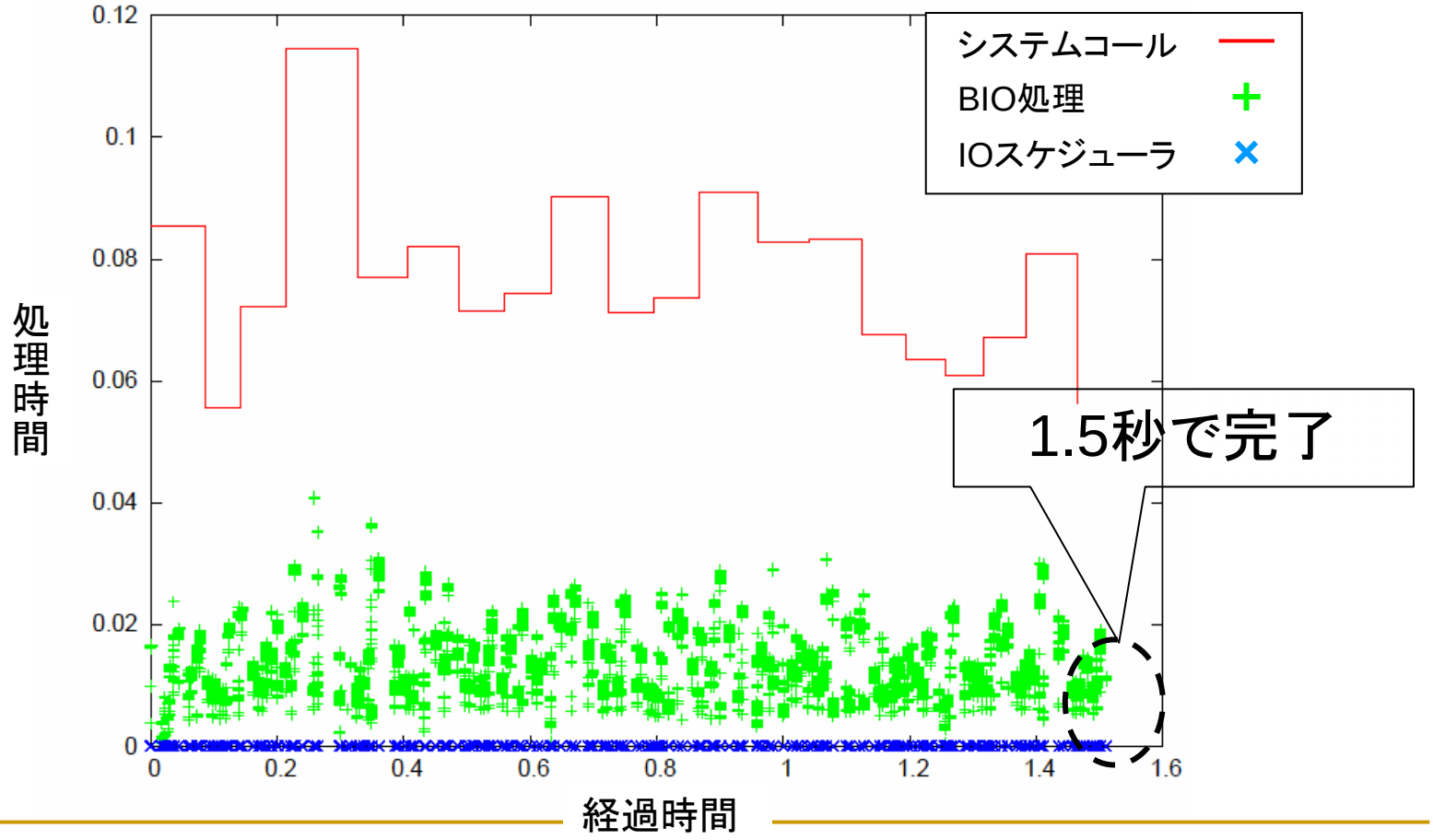
LKSTによる解析(1-2) - パターンC -

- syscallとbiotimeが遅くなる



LKSTによる解析(1-3) - パターンE -

- syscallとbiotimeがかなり不安定かつ遅い



LKSTによる解析(2) – BIO処理時間 –

- どうしてBIO処理時間が増えているのか？
  - ファイルサイズは変わっていない
  - シークが増えた

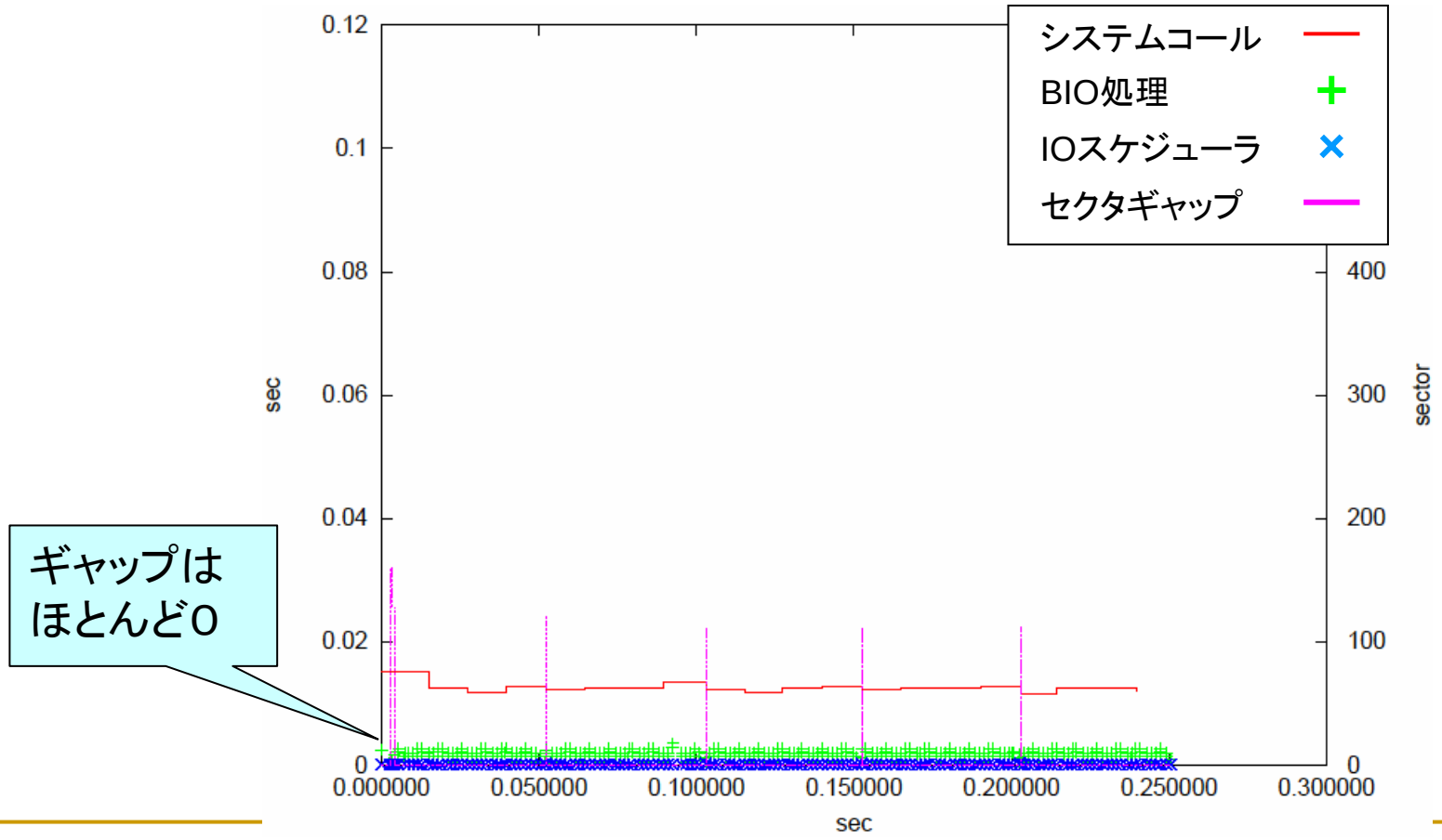
- シーク
  - ブロック間のギャップ

→I/Oセクタのギャップで観測できるはず

LKSTによる解析(3-1)

– IOセクタ差分との関係(パターンA) –

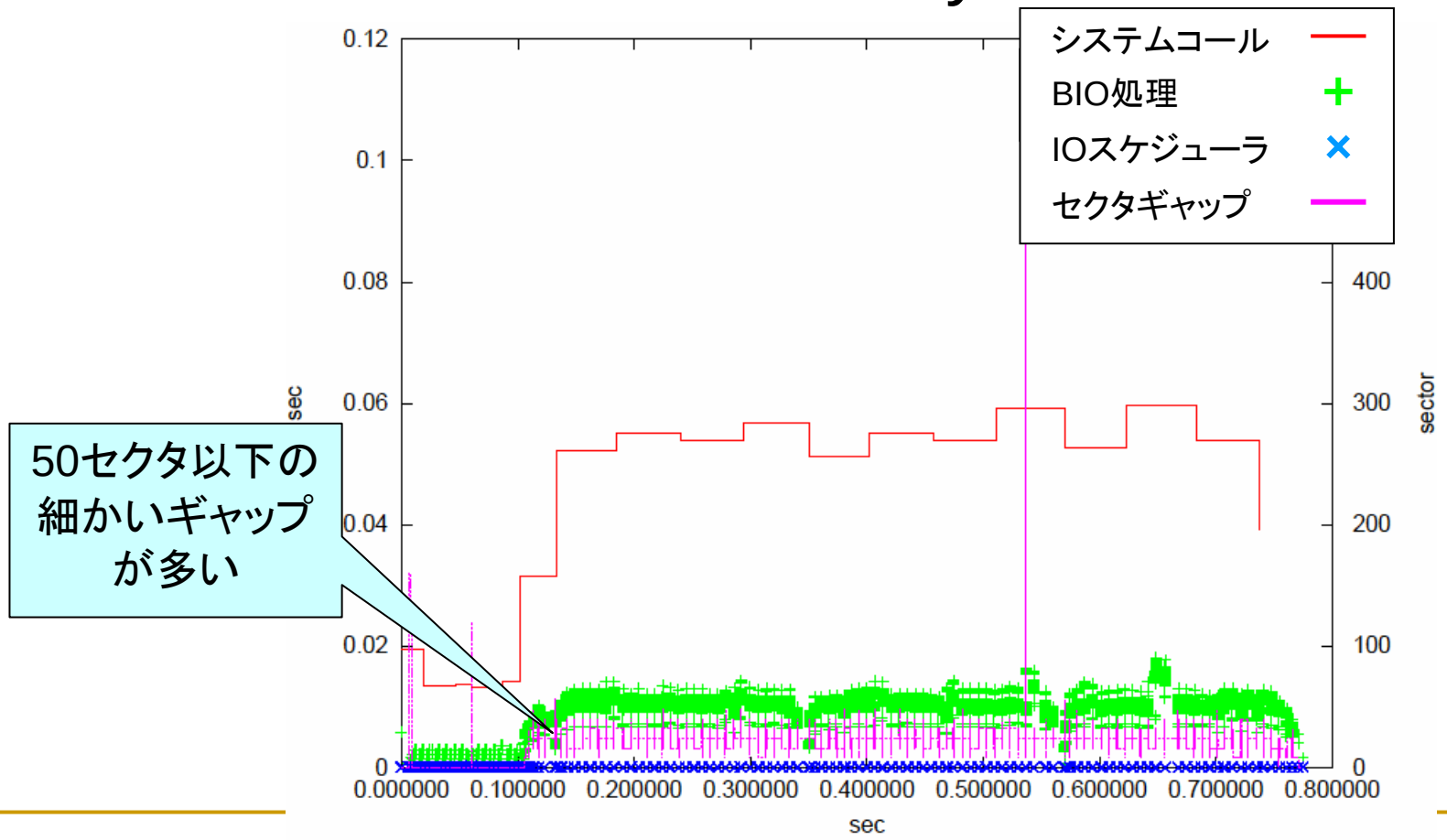
- IOしたセクタ間の差分(ギャップ)を投影



LKSTによる解析(3-2)

– IOセクタ差分との関係（パターンC）–

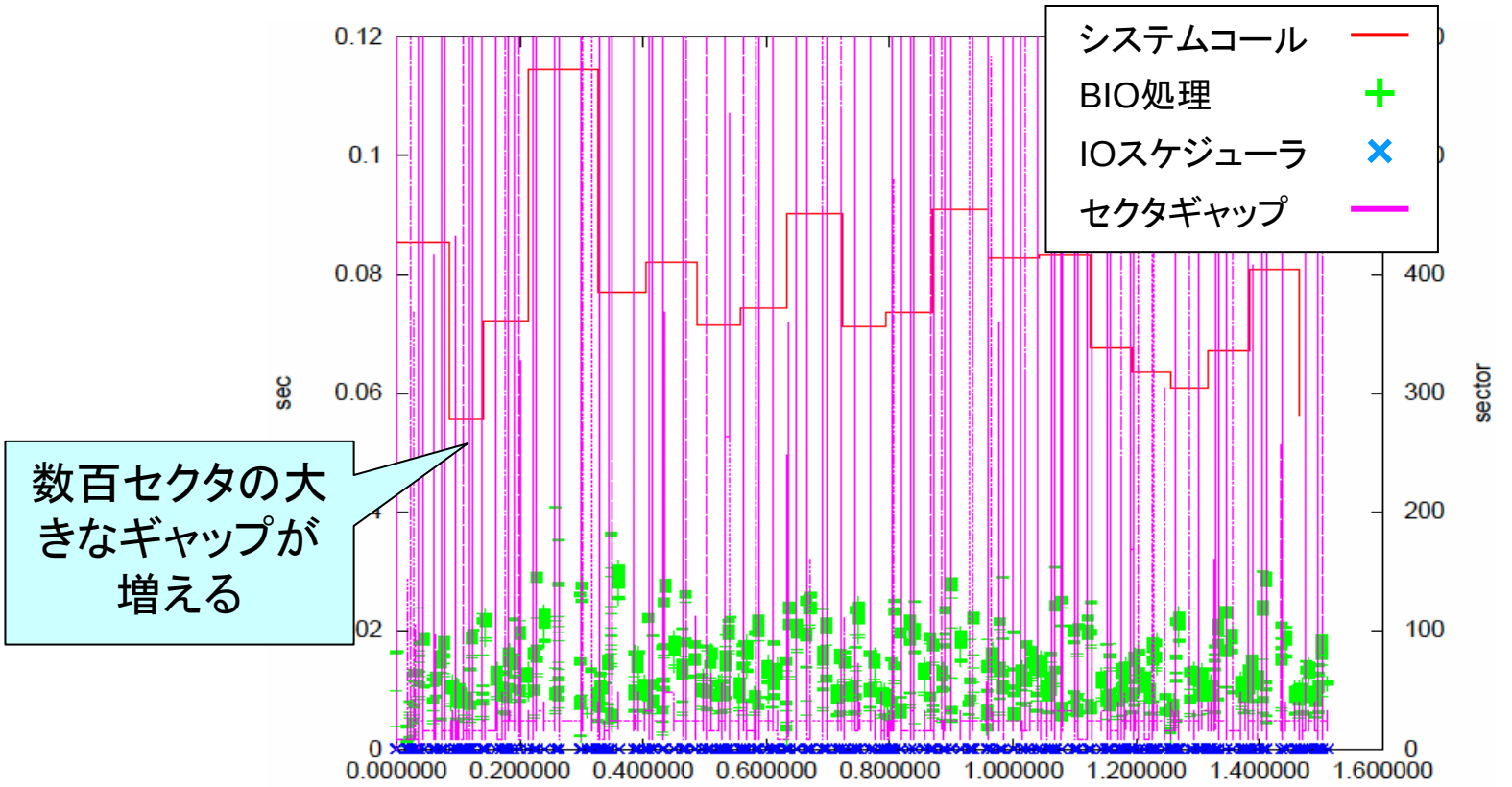
- ギャップが現れるとbiotime・syscallに影響



LKSTによる解析(3-3)

– IOセクタ差分との関係（パターンE）–

- かなり大きなギャップが発生している



ギャップのサイズ・頻度がBIO処理時間に影響

2つのツールのデータの整合性評価（1）

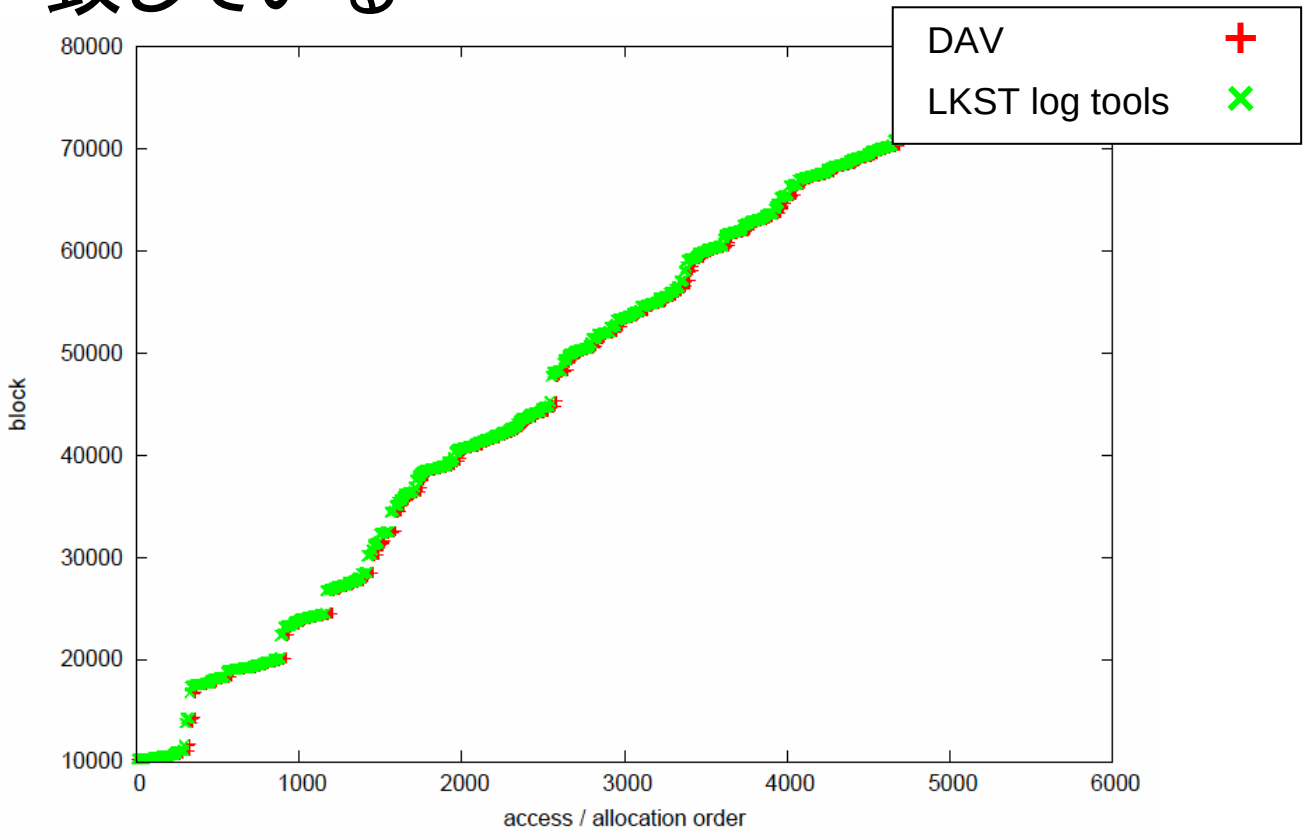
- ギャップの状態は双方で計測できているはず
  - DAVの解析結果
    - ファイルのディスク上でのブロック位置
  - LKSTの解析結果
    - ファイルアクセスしたときのセクタ番号
- 関係式

$$P_B = (P_S - S_{MBR}) / R$$

P_B：ブロック位置
P_S：セクタ位置
S_{MBR}：MBRを含むトラックのセクタ数
R：1ブロックあたりのセクタ数（= 4096/512 = 8）

２つのツールのデータの整合性評価（２）

- ほぼ一致している



DAVのデータからカーネルの処理を予測可能

APPENDIX:

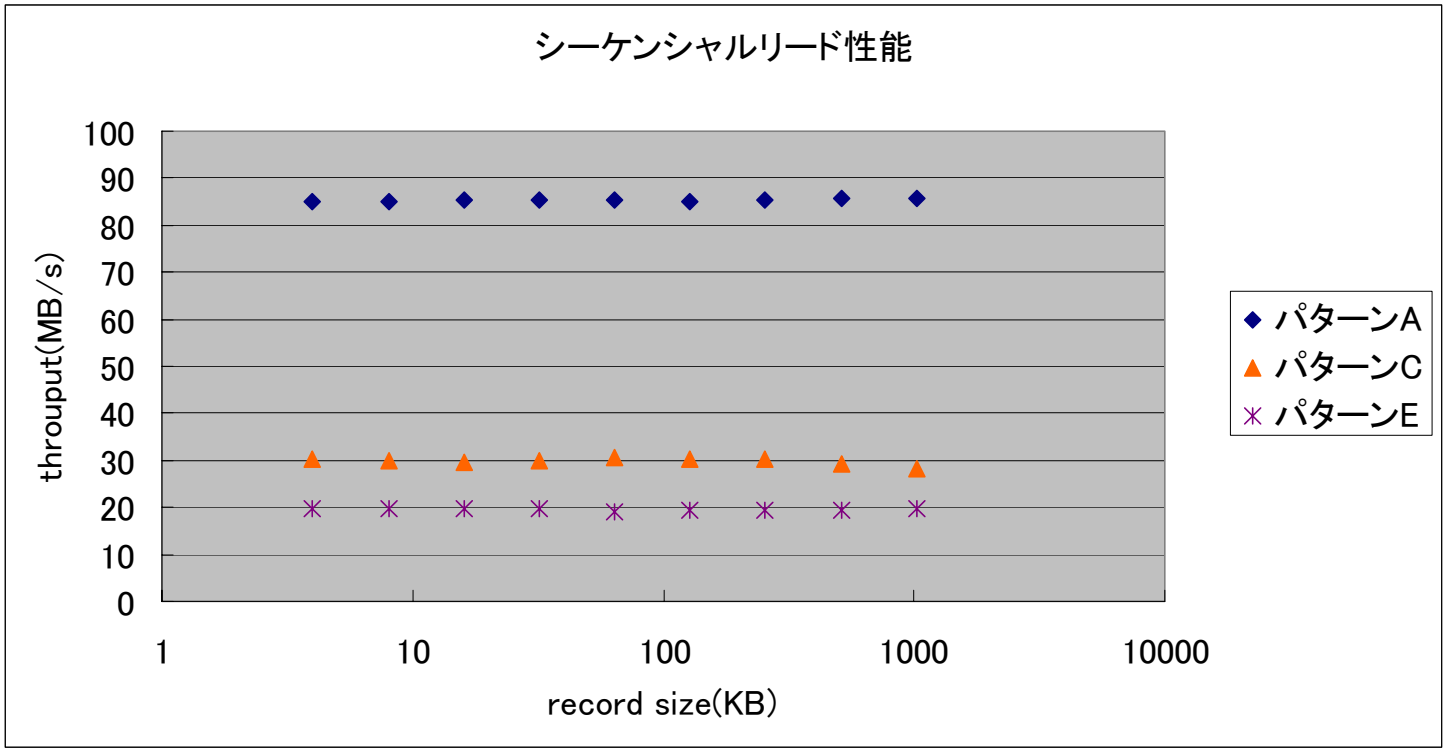
二つのファイルリード順序の比較

影響があるのはシーケンシャルリードだけ?

- 二つのファイルリード順序を評価
  - ランダムリード
  - シーケンシャルリード
- I/Oサイズの変化
  - ファイルシステムのブロックサイズは4KB
  - 4KB～1MBまで変化
- IOzoneを修正
  - 修正前:
    - ランダムリードでは同じ箇所を読む→キャッシュヒット
  - 修正後:
    - ランダムリードで，同じ箇所を読まない.

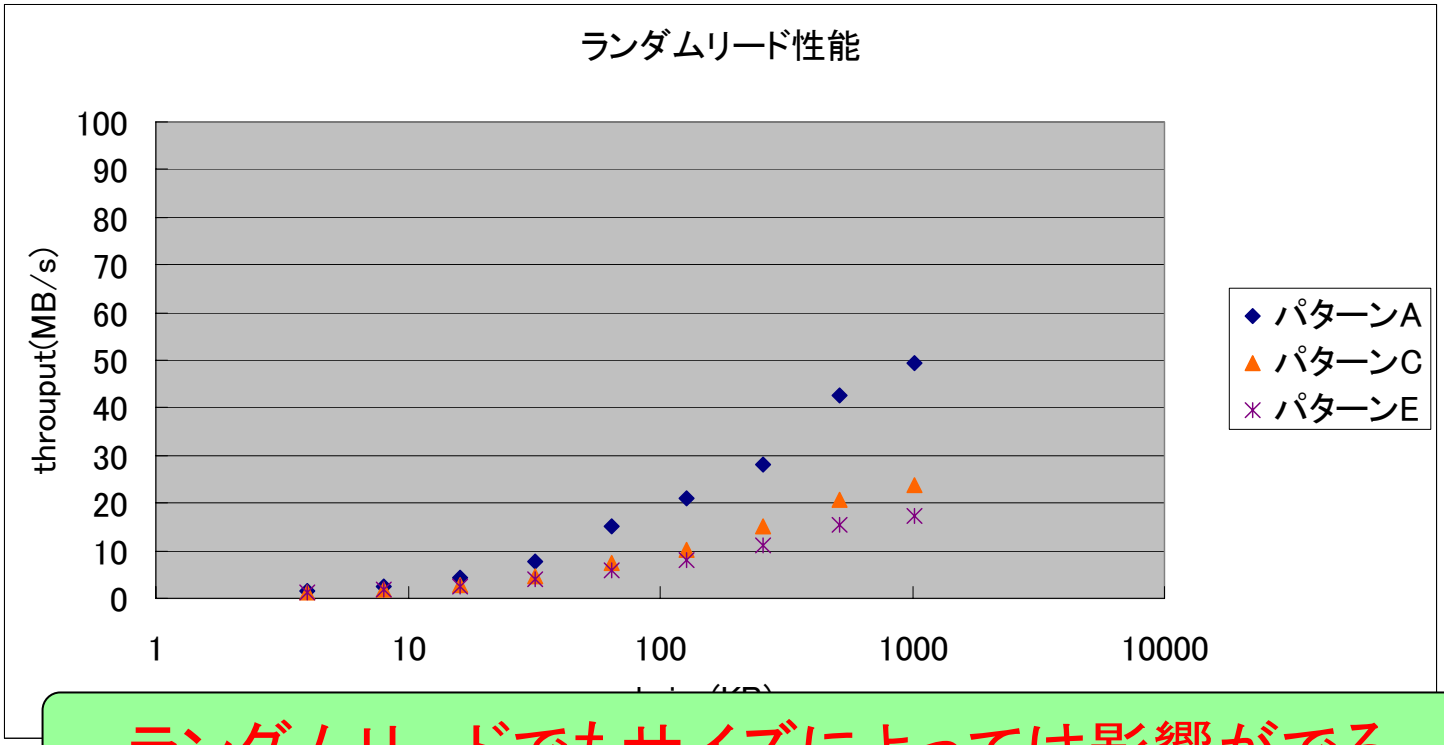
APPENDIX:
シーケンシャルリード

- ほぼ全サイズで安定
  - 断片化無しならば、約85MB/sを維持
- 断片化により性能に3倍近い差が存在



APPENDIX:
ランダムリード

- 徐々に性能が良くなる
  - ただし最大で約50MB/s
- 断片化により32KBあたりから2倍以上の差



ランダムリードでもサイズによっては影響がでる

- はじめに
- ツールの開発
- フラグメンテーション影響の解析
- まとめ

まとめ

- フラグメンテーションの可視化機能の開発
  - DAV (http://davtools.sourceforge.net)
    - 静的なデータブロックのフラグメンテーションを可視化
    - ディスク全体・ディレクトリ・ファイル単位の可視化が可能
  - LKST Log Tools (http://lkst.sourceforge.net)
    - カーネルの処理性能の解析が可能
    - 性能評価対象ごとに個別の性能指標の解析が可能
      - フラグメンテーションによる影響の様子を可視化
  - 機能連携
    - 特定ファイルに対するアクセス性能予測に利用可能
    - 定期的ヘルスチェック⇒DAV
    - 障害解析や原因究明⇒LKST Log Tools

今後の目標

- ツール類
  - デフラグツールの検討
- 性能解析
  - 解析項目の追加（IDE，SAN等／ファイルシステム）
- Linux本体
  - ブロックデバイス層の改良
    - フラグメンテーションが起きても性能劣化し難い方式
  - ファイルシステム層の改良
    - フラグメンテーションを起こし難いブロック配置方式

商標

- 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。
- Intel, Intel Xeon, Pentium は、アメリカ合衆国およびその他の国におけるインテル コーポレーションまたはその子会社の商標または登録商標です。
- その他の記載されている社名および製品名は各社の登録商標また商標です。

謝辞

- DAVとLKSTの開発の一部は「独立行政法人 情報処理推進機構 オープンソースソフトウェア活用基盤整備事業」に係る委託業務の調査・開発に基づくものです。ご支援いただいたことに感謝いたします.

