

その他の ファイルシステム



今回は
新しめのファイルシステム
を考えます



今回は 新しめのファイルシステム を考えます

それまで使われてきたファイルシステムの
不便な点を直すような新機構が提案され
既存OS (Win, Linux等) に取り入れられる



比較的新しいファイルシステム

1. ジャーナル (ジャーナリング) ファイルシステム 教7.9

- 書換えの記録を保存
 - ⇒ 回復可能で耐故障性を高める利点
もともとデータベースで考えられた技術
- Windows (NTFS) やLinux (Ext3) で利用

2. ログ構造化ファイルシステム 教7.8

- 書換え分を蓄積する形
 - ⇒ 書いた場所が散らばらない~アクセス高速



ジャーナリングファイルシステム の考え方 (教7.9.1参照)

- ファイルシステムの更新の時に
途中で中断・電源断・クラッシュがあると
内部のデータ (⇒次頁) に不統一発生
⇒ 読めなくなる (回復できなくなる)
- それを防ぐために、
一連の操作を「トランザクション」と
して扱う



4

ジャーナリングファイルシステム の考え方 (教7.9.1参照)

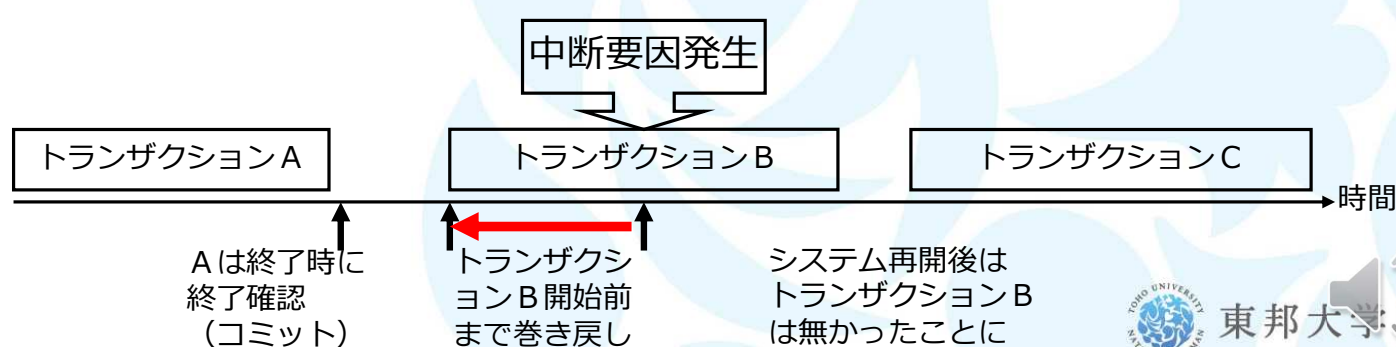
- 不統一になると困る内部のデータの例
 - iノードで、一連のリンクを書換えている途中で、
一部が書換え済み、一部が書換え前状態が残る
- トランザクションとは
 - データベースなどで用いられる用語で、
一連の操作をまとめたもの (不可分操作に類似)
途中の状態を許さず、開始前か終了後の
どちらかの状態しかないようにした操作



5

トランザクションのイメージ

- この間に中断されると管理データの不統一が発生し後から継続できないなどのトラブルになる
- 終了させるか、どうしても中断する場合は開始前の状態に巻き戻す
 - (トランザクションが無かったことにする)
- 詳しくはデータベースなど他の授業で



6



東邦大学

ジャーナリングファイルの動作

- データ本体の変更 (量が多いと想定) より前に管理データ (i-nodeなど) の変更分をログ領域 (ジャーナル) に記録する
- ジャーナルは定期的に、ファイル (本番) システムに書き戻し (コミット) てクリアする
- 前のコミットから現在 (クラッシュ時) までの管理データの変更を回復できる
 - ⇒ より確実かつ高速に回復できる
- 様々な拡張・改良が提案されている
 - 例: 管理データ+データ自体もログ (⇒重い)

7



東邦大学

ジャーナリングファイルシステムの例

- 最初はIBM AIX上でのJFSらしい（1990）
 - 目的はフォルトトレラント（障害発生時にもシステム全体は停止せず、動作を継続できるシステム）
- Linux上のext3, ext4ファイルシステム
 - ext3はそれまでのi-ノードファイルシステムext2と互換のままジャーナリングを取り入れたもの
 - ext4はext3同様ジャーナリングを利用するが、更に連続物理ブロックにより大きなファイルの効率を向上
- Windows NTFS

8

ログ構造ファイルシステム

- もはや、ログしかない
データも i-ノードも、一切の変更をログに記録
- ねらい： ファイルシステムの性能向上
 - ディスクは大きなメモリにキャッシュされる
⇒ ディスクには殆ど書込みだけ
⇒ 変更だけを後ろへ後ろへ追加するとディスクの場所が限られる ⇒ シークが減る
 - 勿論、ジャーナリングシステムの耐障害性も持つ

9

ログ構造ファイルシステム

- Unix/Linux系での実装
 - Sprite OSで最初の実装 (Rosenblum, 1992)
Rosenblum, Ousterhout: "The Design and Implementation of a Log-Structured File System".
ACM Transactions on Computer Systems, Vol.10, No.1, February 1992.
 - NILFS (NTT) はLinuxにマージされた (天海, 2005)
<http://www.nilfs.org/ja/>
天海他: "Linux 用ログ構造化ファイルシステム nilfs の設計と実装". 第 99 回システムソフトウェアとオペレーティング・システム研究会, 情報処理学会, 2005 年 5 月.
- フラッシュメモリでのファイルシステムとして利用
 - JFFS
 - YAFFS

10



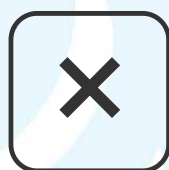
ここまでのまとめ

- ファイルシステムの改良は現在も進行中
 - 使いにくさの克服
 - 障害時のデータ欠落や統一の回復
 - 最近の環境に合せた性能向上策
 - 大きなメモリ ⇒ キャッシュ効果大 ⇒ writeが多い
 - ⇒ 更新部分を固めてシークを減らす
 - ディスク容量が大きい、ファイルが大きい・多い
 - フラッシュメモリなど新しいデバイス
 - ⇒ 書換え回数の上限があるので減らす工夫必要

11



ファイルシステムが
現在も改良され続けていることが
理解できましたか？



↓
次へ

