ext3諸元拡大に関する研究開発動向、及び改造方式の検討

平成18年6月2日

NECソフトウェア東北株式会社
佐藤　尚

NEC Empowered by Innovation

NEC Software Tohoku, Ltd.

目次

第1章　ext3ファイルシステムの概要

1-1　ext3ファイルシステムとは？

1-2　ext3のデータ構造

NEC

Empowered by Innovation

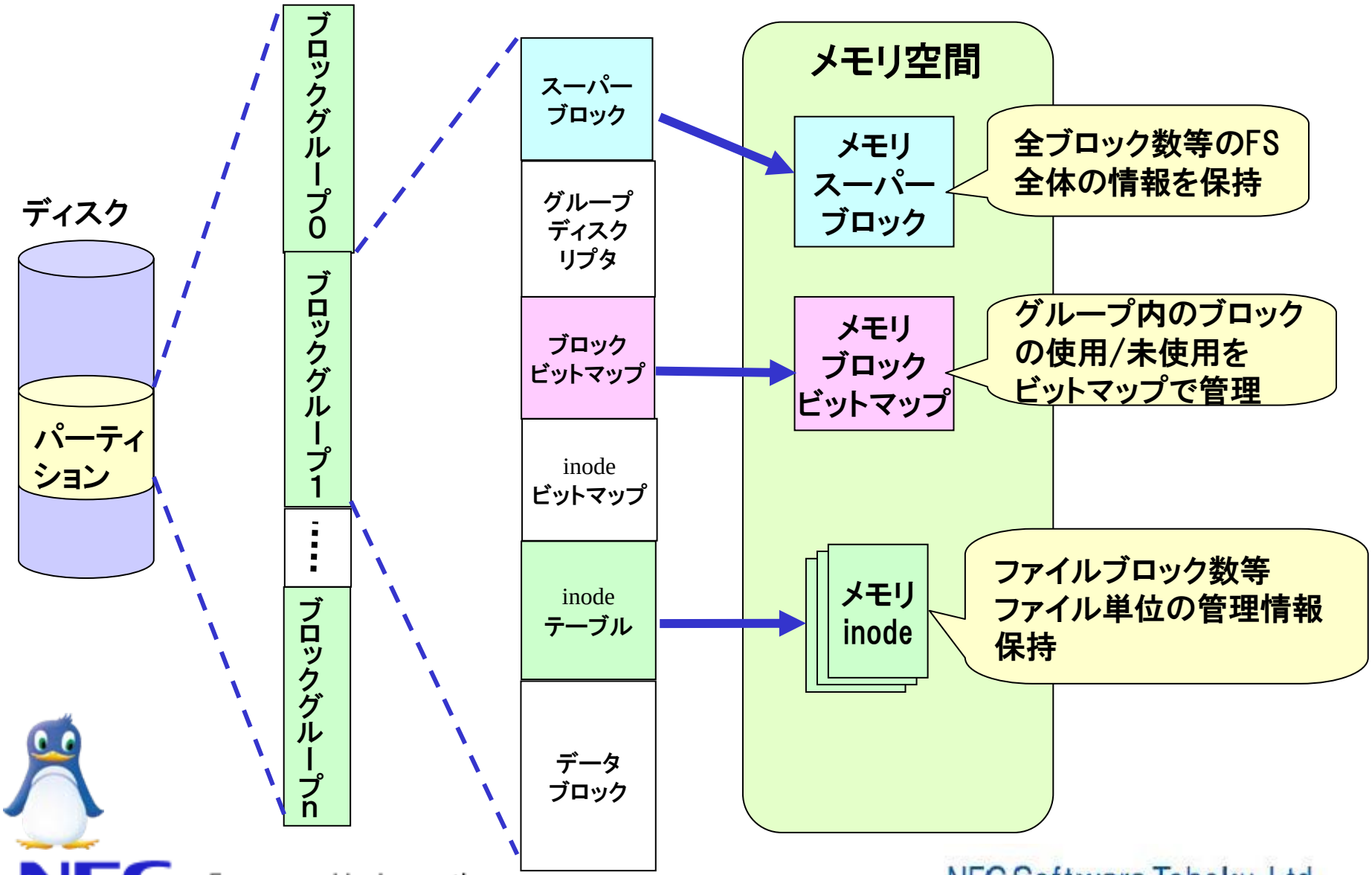
NEC Software Tohoku, Ltd.

1-1 ext3ファイルシステムとは？

Linuxのネイティブなファイルシステムである
ext2にジャーナル機能を追加したファイル
システムである。

RHEL4等の多くのディストリビューションにおいて標準
ファイルシステムとして採用
【利用実績が多く、品質が安定している】

1-2 ext3のデータ構造



第2章　ext3の諸元に関する問題点

2-1　他ファイルシステムとの比較

2-2　ディスクストレージ容量の増大

2-1 他ファイルシステムとの比較

Linuxの主要ファイルシステムの諸元

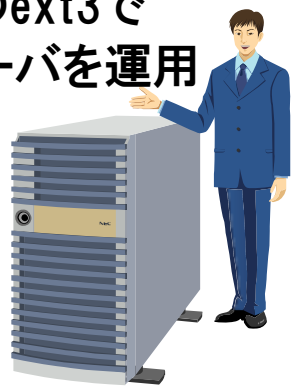
FS種別	OS種別	32bit環境		64bit環境	
		最大FSサイズ	最大ファイルサイズ	最大FSサイズ	最大ファイルサイズ
XFS	Linux	16TB	16TB	8EB	8EB
JFS	Linux	16TB	16TB	32PB	4PB
ReiserFS	Linux	16TB	2TB	1EB	16TB
ext3	**Linux**	**8TB**	**2TB**	**8TB**	**2TB**

ext3の諸元は他ファイルシステムに比べて小さい。

2-2 ディスクストレージ容量の増大

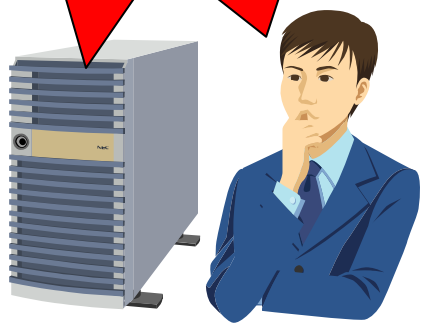
過去2年間でディスクストレージの出荷容量は約2倍に増大。
（調査機関調べ）
このペースが継続すると・・・

現状3TBのext3で
ファイルサーバを運用



4年後

ディスクが12TBとなり、
ext3で運用できない！



第3章　ext3の諸元を制限している要素

3-1　ファイルシステムサイズの制限

3-2　ファイルサイズの制限

3-1 ファイルシステムサイズの制限

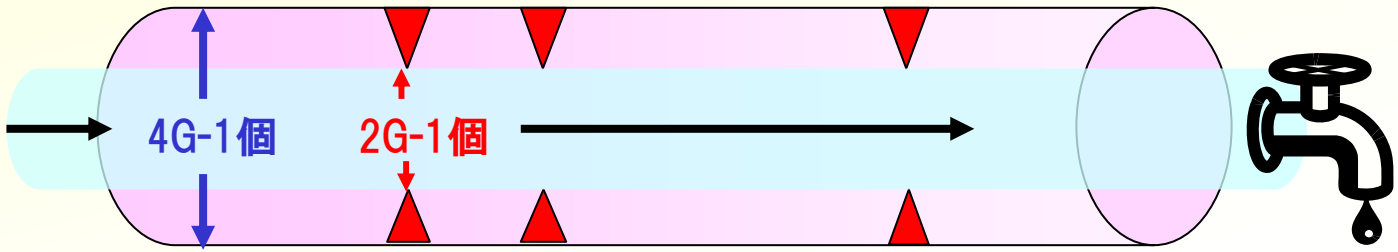
ファイルシステムブロック数の制限

```
int ext3_new_block(handle_t *handle, struct
                        inode *inode,
                   unsigned long goal, int *errp)
{       :       :
int group_no;
int goal_group;
int ret_block;
```

ブロック番号、ブロック数を符号付の4バイトで宣言

同類箇所
約100箇所

ブロック数を保持する構造体エントリは符号無し4バイト(最大2^{32}-1 (4G-1)まで保持可能)なのに、最大ブロック数が2^{31}-1(2G-1)個に制限されてしまう。



3-1 ファイルシステムサイズの制限

ファイルシステムサイズの制限

最大ブロック数 ＝ $2^{31} - 1$
＝ 2G – 1個

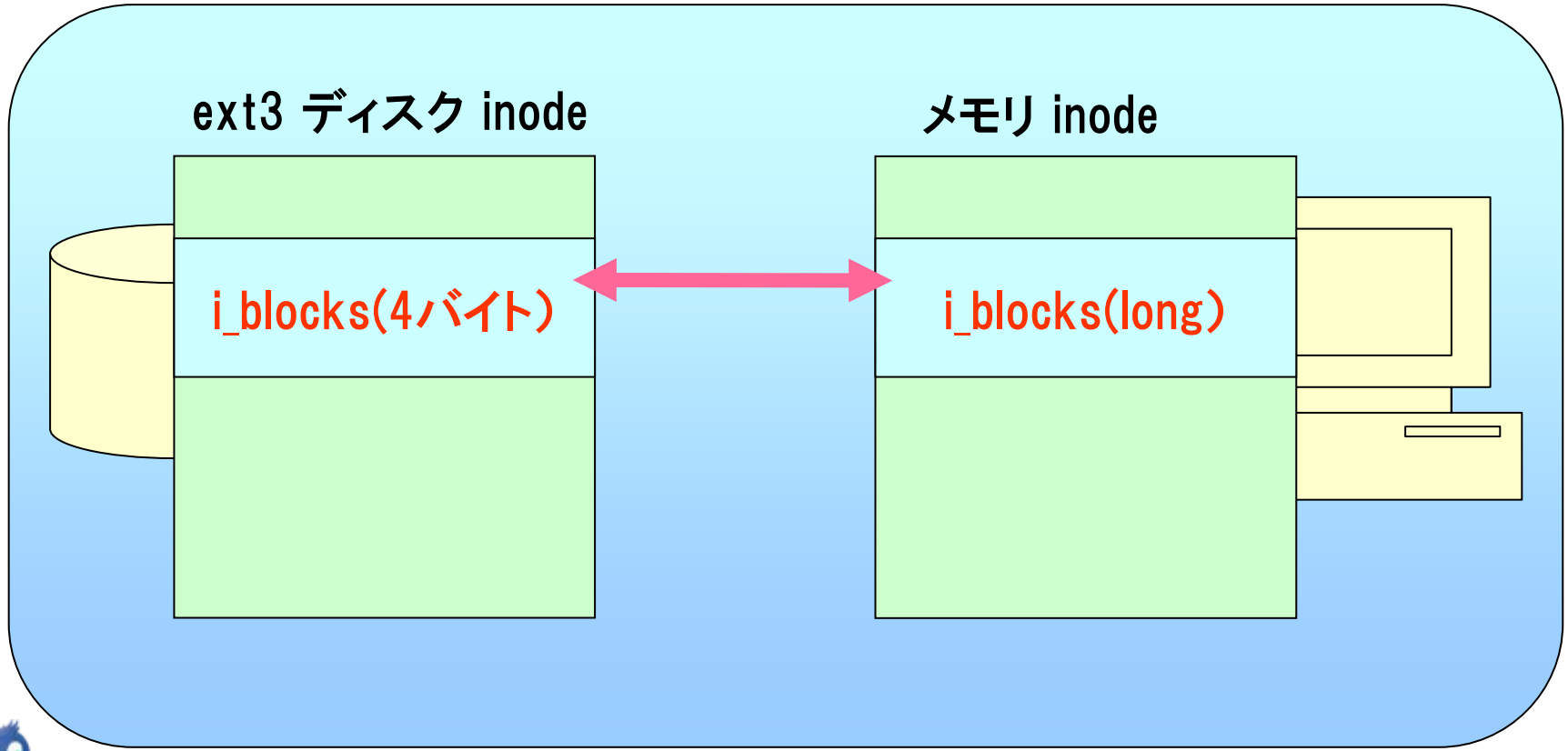
最大ブロックサイズ ＝ 4096バイト

（但し、制限を外せばページサイズまでのブロックサイズを使用可能）

(2G -1) × 4096 ≒ 8Tバイト

3-2 ファイルサイズの制限

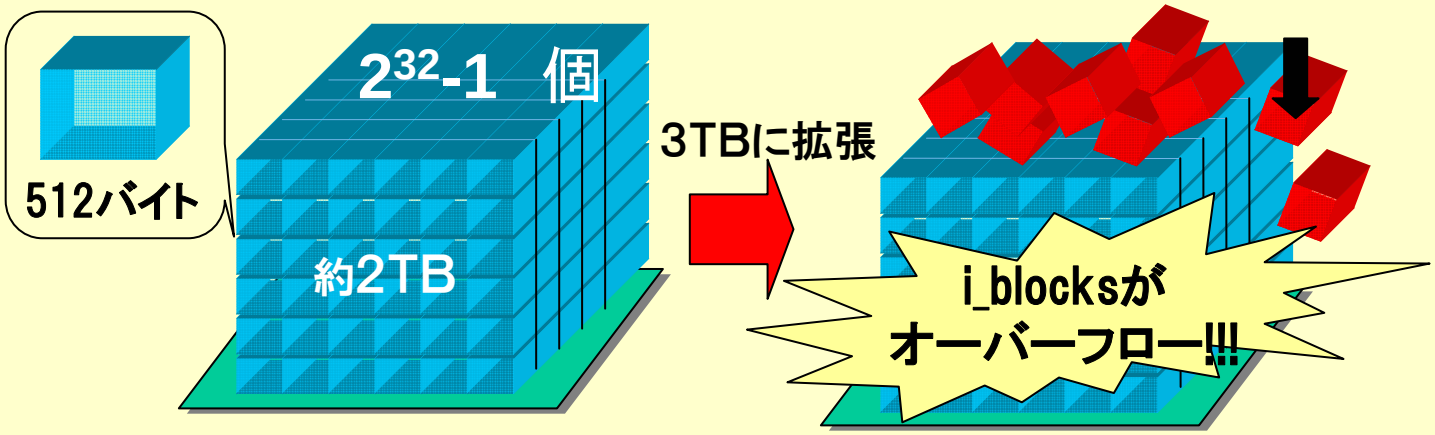
- 512バイト単位のブロック数を保持する(i_blocks)による制限



3-2 ファイルサイズの制限

512バイト単位のブロック数を保持する(i_blocks)による制限

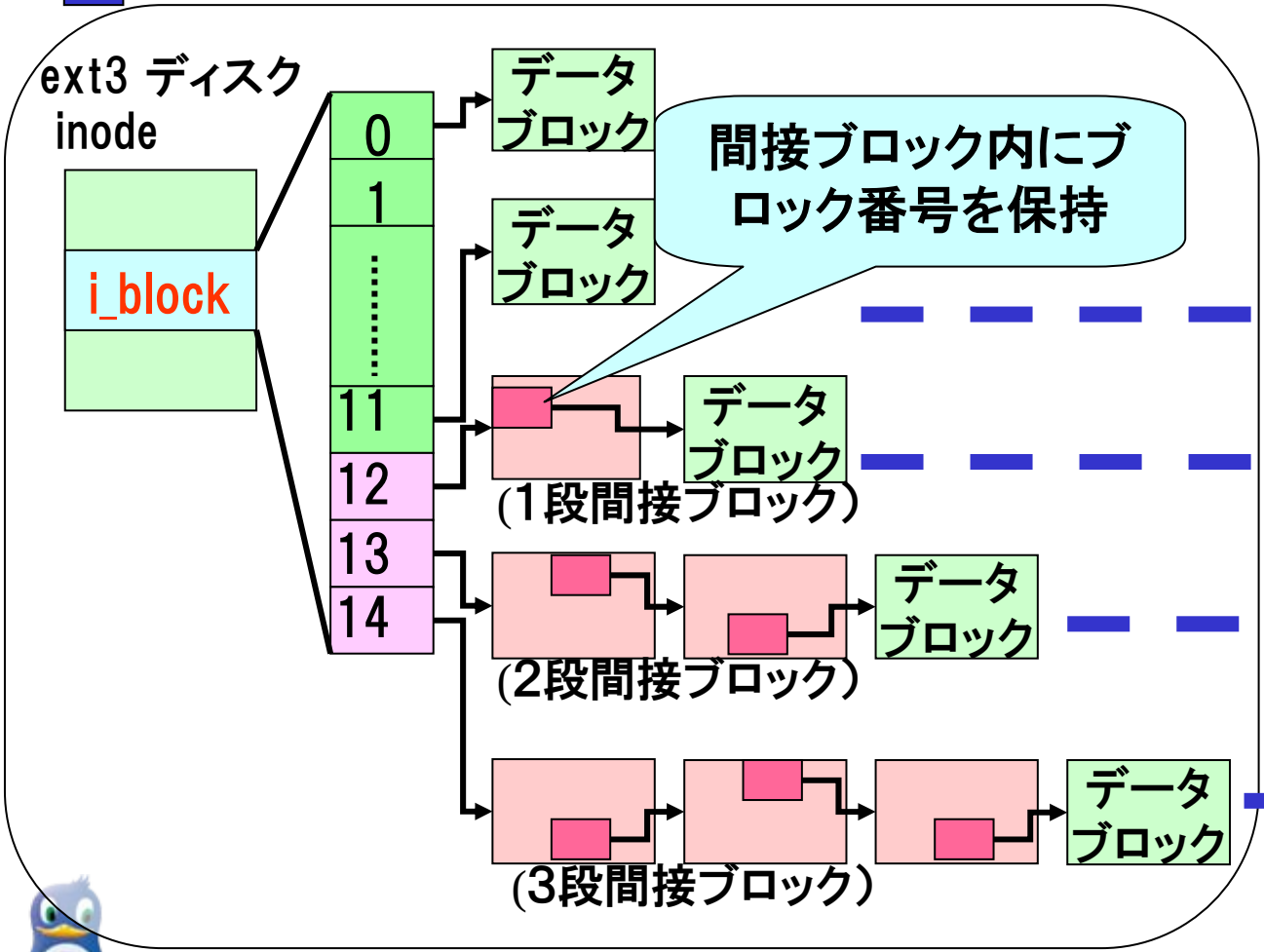
- ディスクinodeのi_blocksによる制限
  最大ファイルサイズ＝512 ×(2^{32}－1) ≒ 2Tバイト に制限。



- メモリinodeのi_blocksによる制限
  32bit環境では、longは4バイトのため最大ファイルサイズが
  約 2Tバイト に制限。
  （※全ファイルシステムで、2Tバイト以上のファイルにおける
  i_blocksが不正となる）

3-2 ファイルサイズの制限

間接ブロック方式のファイルブロック管理による制限



最大ファイルサイズ＝
(
12
+1024
+1024²
+1024³
)
×4096(最大ブロックサイズ)

≒ 4.1TB に制限

第4章　既存のext3諸元拡大方式

4-1　Goldwyn Rodrigues氏の方式

4-2　Laurent Vivier氏の方式

NEC

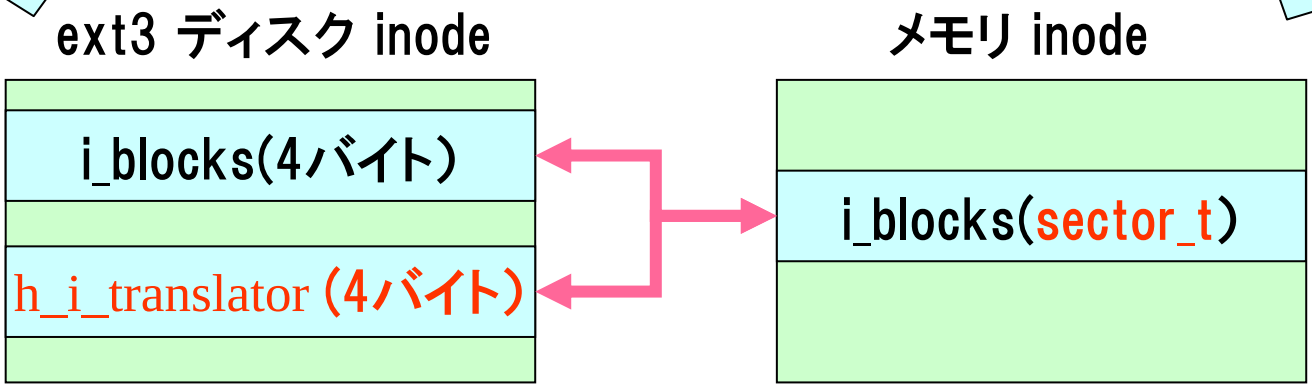
Empowered by Innovation

NEC Software Tohoku, Ltd.

4-1-(1) Goldwyn Rodrigues氏の方式

h_i_translatorを
i_blocksの上位4バイトとして使用

i_blocksの型をsector_t
(通常システムでは8バイト)に変更



	改造前	改造後
最大ファイルサイズ	2TB	4.1TB

4-1-(2) 長所、及び短所

長所

- 少ない修正量でファイルサイズを拡大できる。

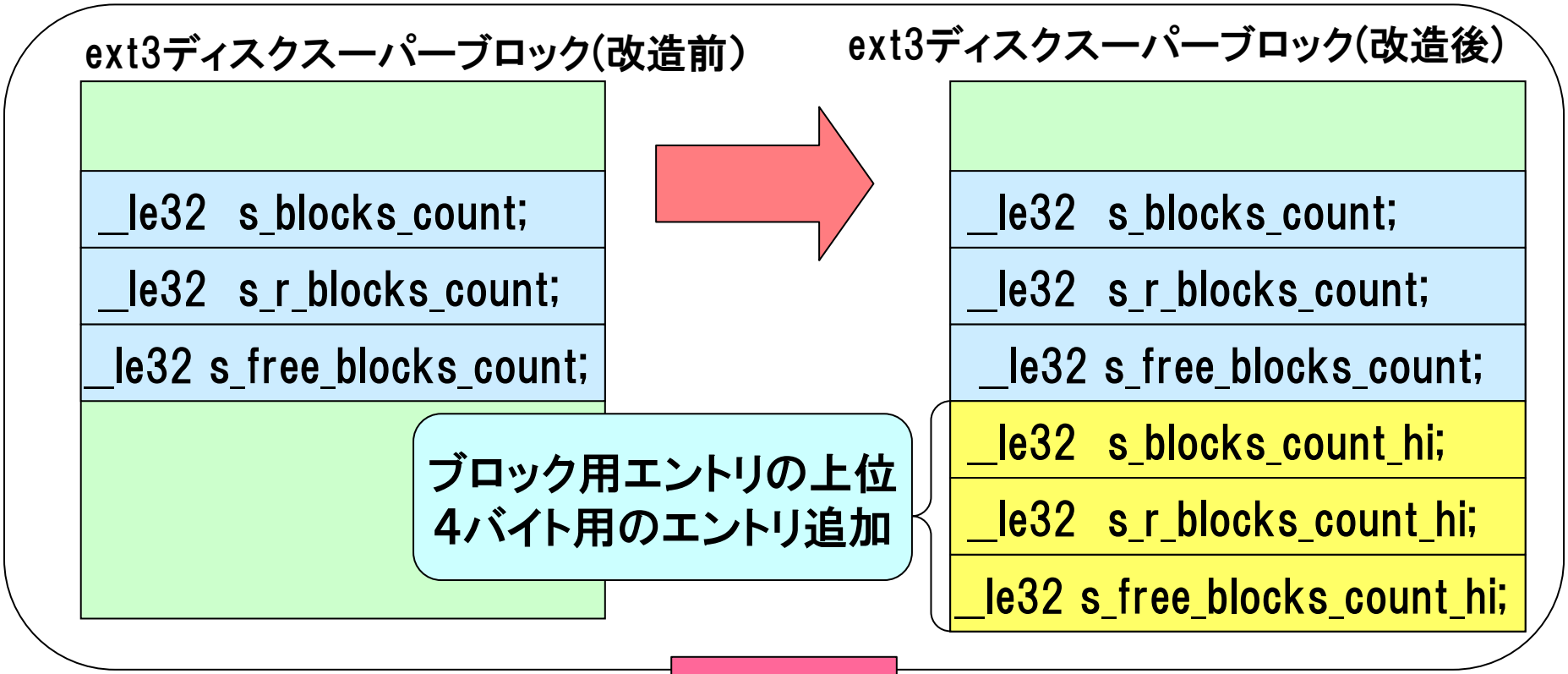
短所

- ファイルシステムサイズを拡大できない。

- h_i_translatorを使用しているため、HURDでext3を使用できない。

4-2-(1) Laurent Vivier氏の方式

ブロック関連の構造体エントリ、及び変数の8バイト化



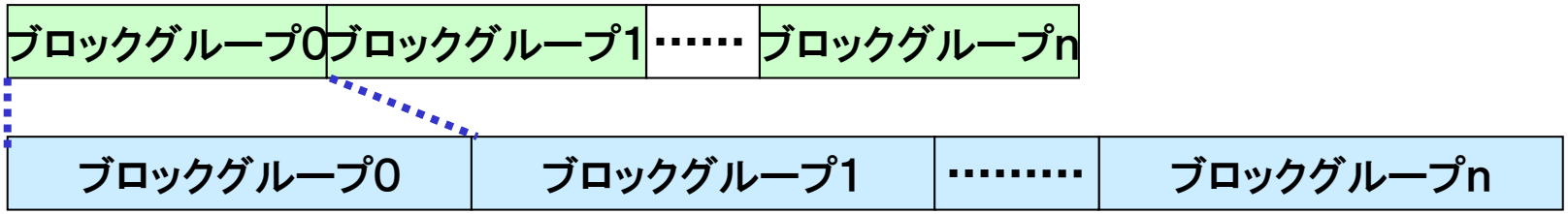
2^{32} 個以上のブロック数保持可能

4-2-(1) Laurent Vivier氏の方式

その他の性能向上のための改造

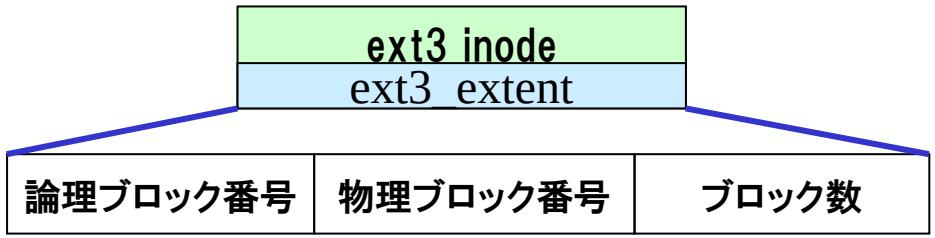
ブロックグループ内の最大ブロック数増大

連続ブロック増大によるI/O性能向上



エクステントによるファイルブロック管理

連続ブロックを1つのエクステントで管理することにより
ブロック検索性能向上



4-2-(2) 長所、及び短所

長所

- 最大ファイルシステムサイズ、最大ファイルサイズ共に大幅な拡大が可能。

	改造前	改造後
最大ファイルシステムサイズ	8TB	256PB
最大ファイルサイズ	2TB	256PB

短所

- ディスクフォーマットを変更するため、**下位互換が失われる。**
- 制御ロジックの大規模な修正のため、**品質安定化までに長時間を要する。**

第5章　新しい諸元拡大方式の検討

5-1　改造の方針

5-2　ファイルサイズ拡大

5-3　ファイルシステムサイズ拡大

5-4　ブロックサイズ拡大

5-1改造の方針

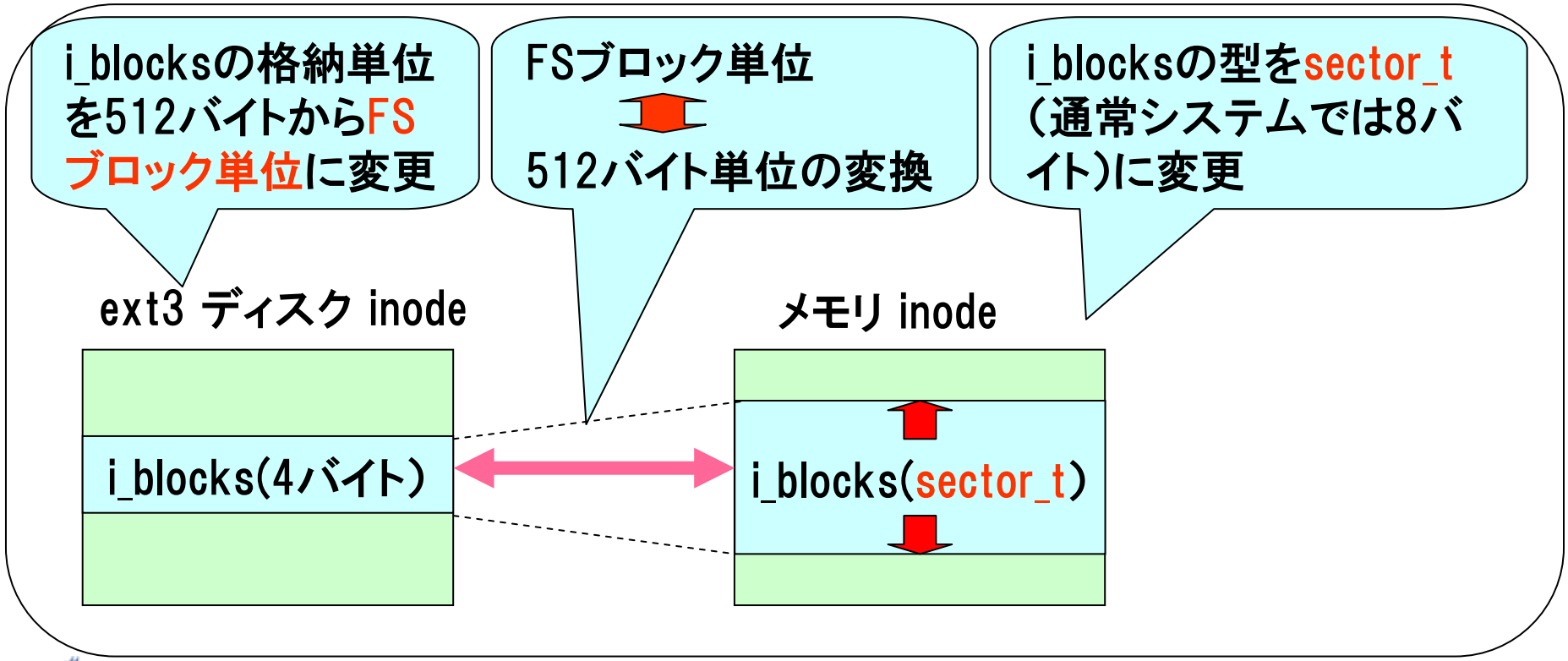
少ない改造量で最大ファイルシステムサイズ、最大ファイルサイズの両方を拡大

制御ロジックの修正を極力最小化し、現状の安定性を維持

ディスクフォーマットを変更せず可能な限り下位互換を維持

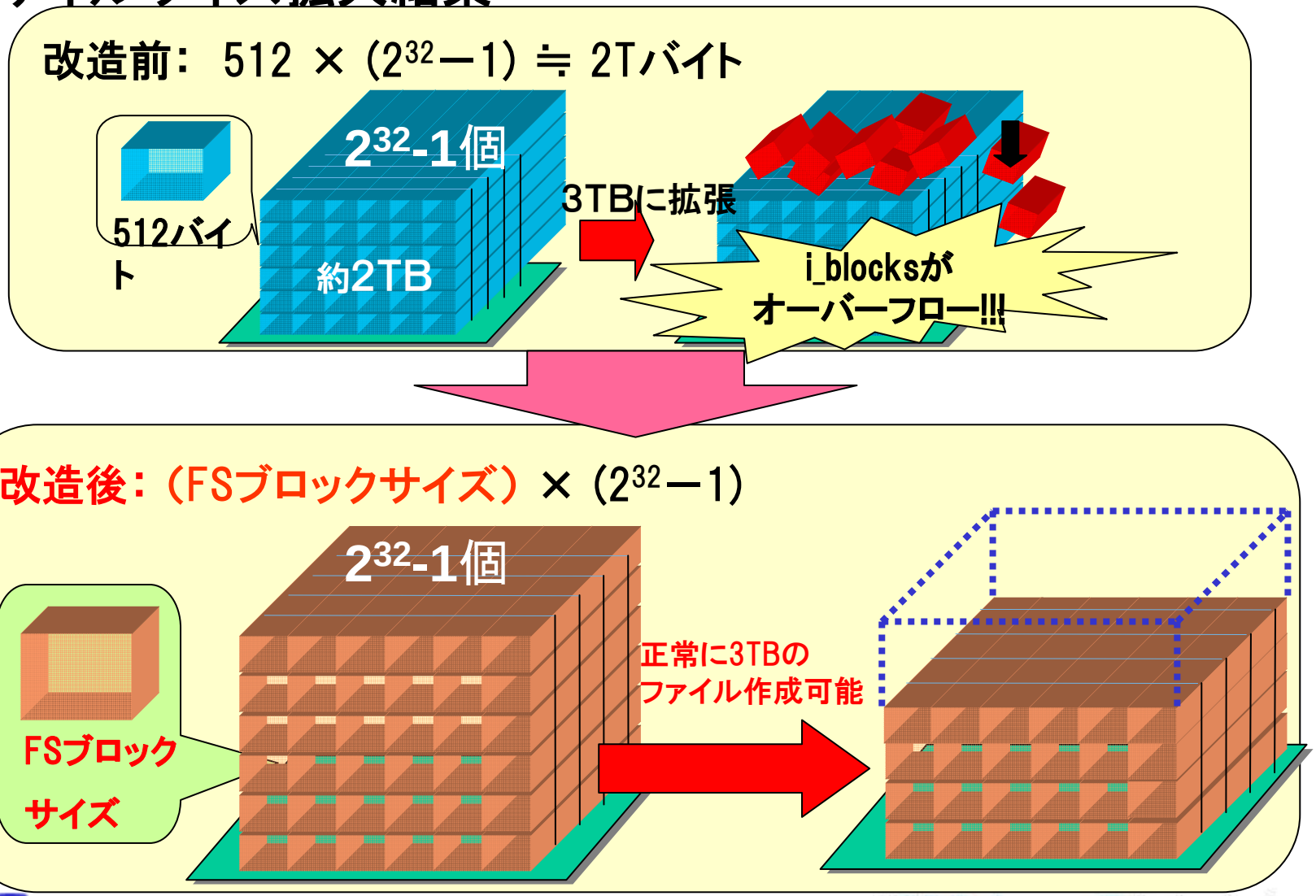
5-2-(1) ファイルサイズ拡大方式

ファイルサイズ拡大方式



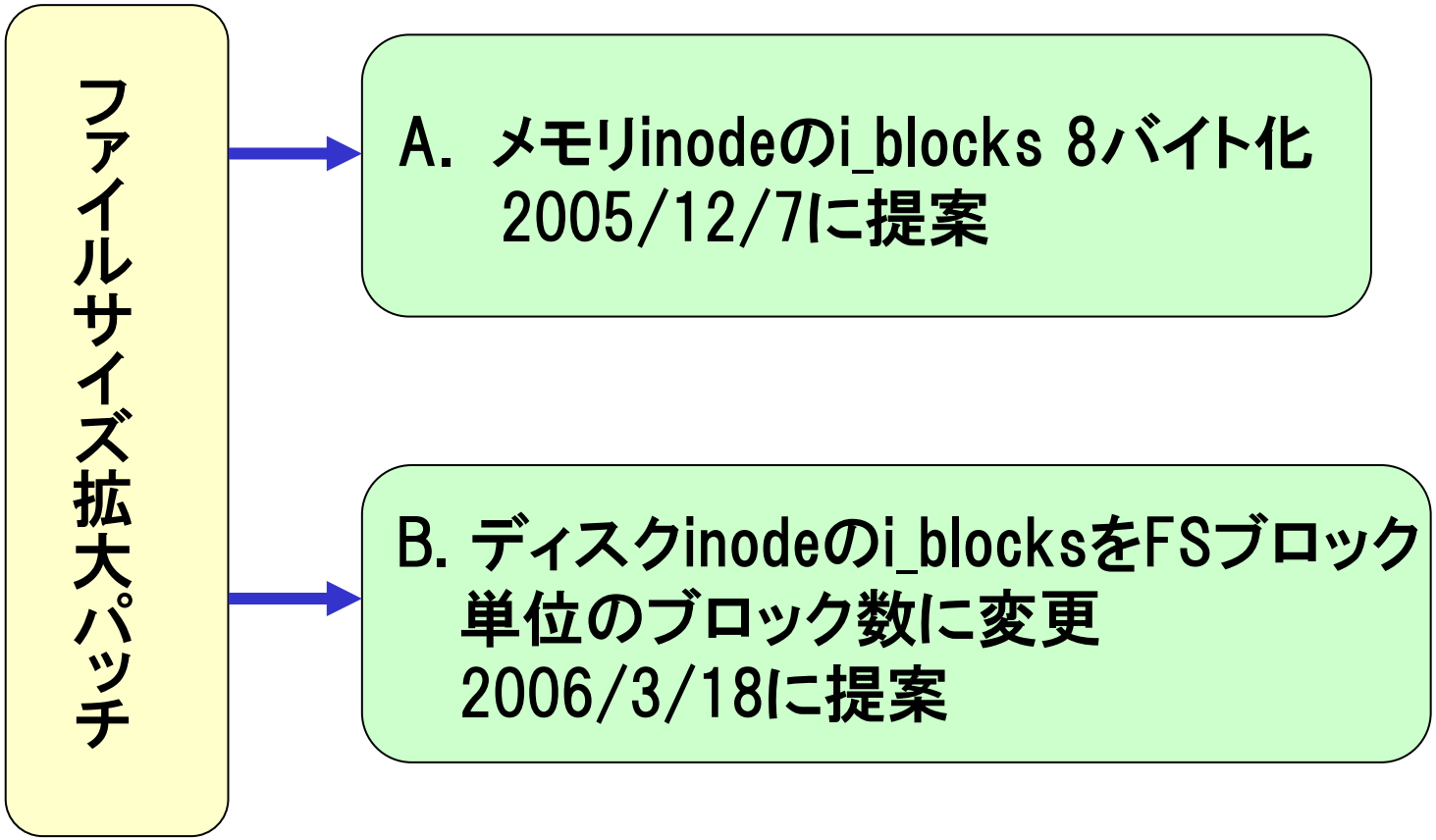
5-2-(1) ファイルサイズ拡大方式

ファイルサイズ拡大結果



5-2-(2) コミュニティへの提案

修正を2つのパッチセットに分割し、ext3開発用メーリングリストに提案。



5-2-(3) パッチAの動向

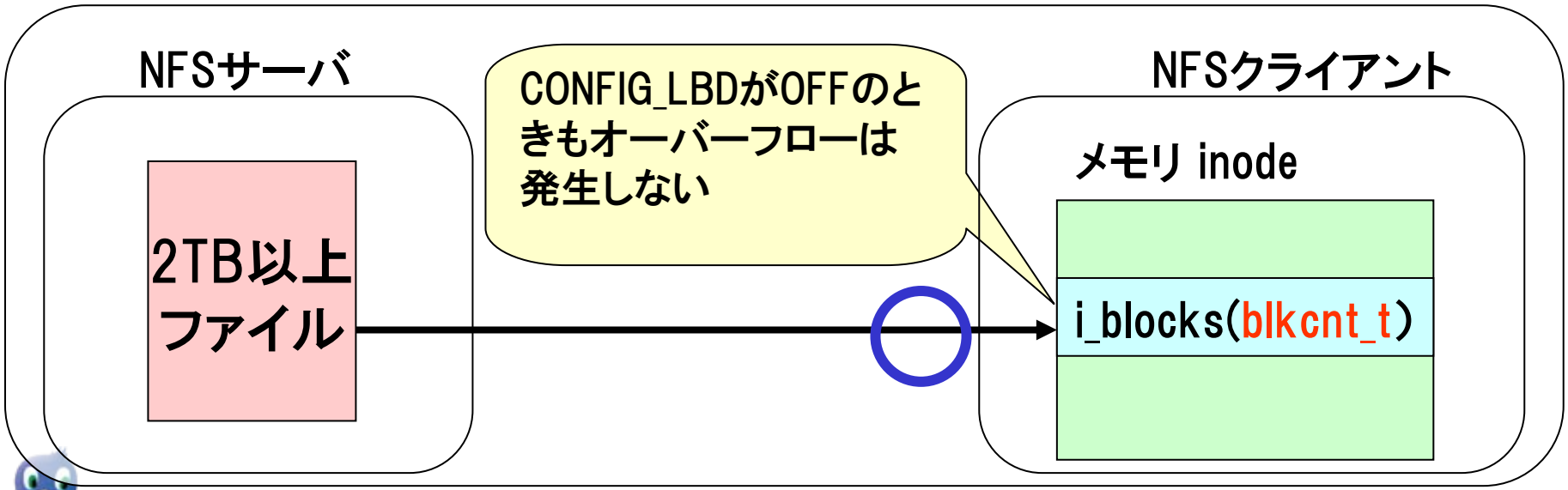
2005/12/7 Trond Myklebust氏からの指摘内容

CONFIG_LBDをOFFにした場合、NFS経由で
2TB以上のファイルにアクセス不可

5-2-(3) パッチAの動向

Trond Myklebust氏からの指摘による修正

CONFIG_LBDとコンフィグレーションを分離するため
ファイルブロック数用の型(blkcnt_t)を新設



5-2-(3) パッチAの動向

Linuxへのパッチ組込み

2006/1/13 Linux 2.6のメンテナAndrew Morton氏により、mm-tree(Linuxの評価用ツリー)に登録。

2006/3/27 2.6.17-rc1において正式にLinuxに採用。

5-2-(4) パッチBの動向

- 2006/3/20 Andreas Dilger氏からの指摘内容

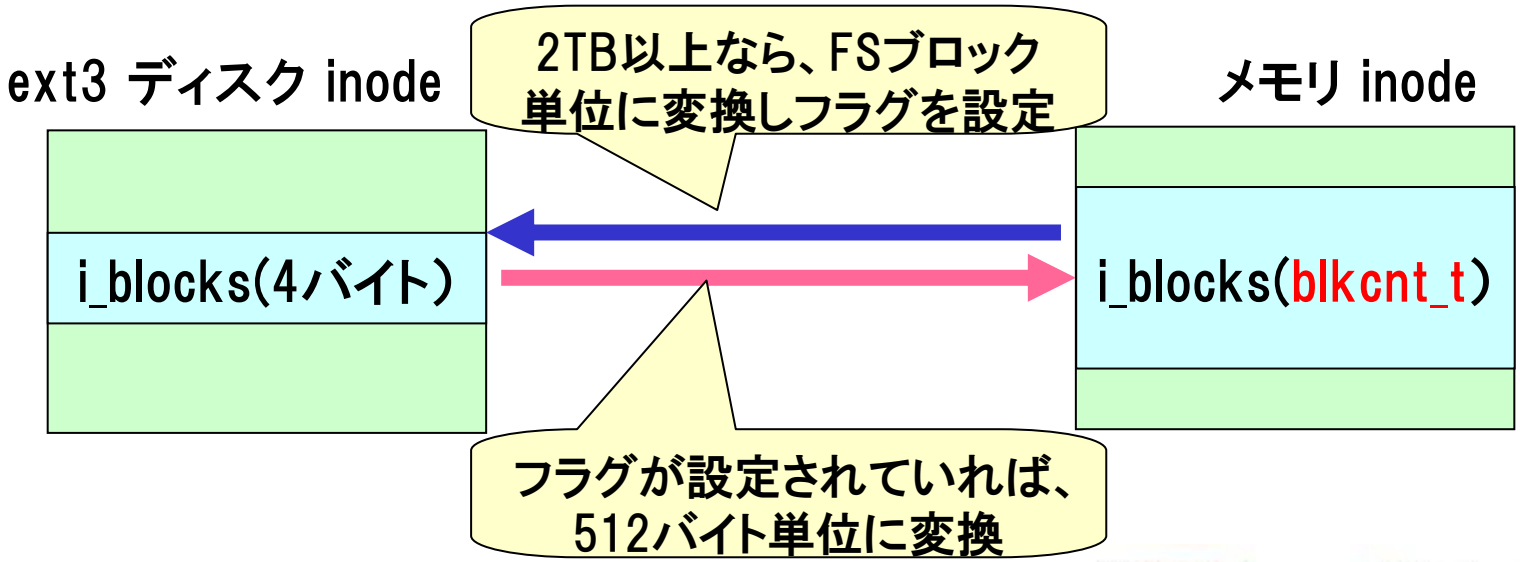
互換性維持のため2TB以上のファイルのみディスクinodeのi_blocksをFSブロック単位として格納すべき。

2TB以上のファイルを作成しない場合は**既存カーネルでも改造後のext3が使用可能**となる。

5-2-(4) パッチBの動向

Andreas Dilger氏からの指摘による修正

- inode書き込み時、ファイルサイズが2TB以上の場合はi_blocksをFSブロック単位で格納。またディスクinodeにフラグ(EXT3_HUGE_FILE_FL)を設定。
- inode読み取り時、EXT3_HUGE_FILE_FLが設定されているファイルのみ、FSブロック単位から512バイト単位に変換



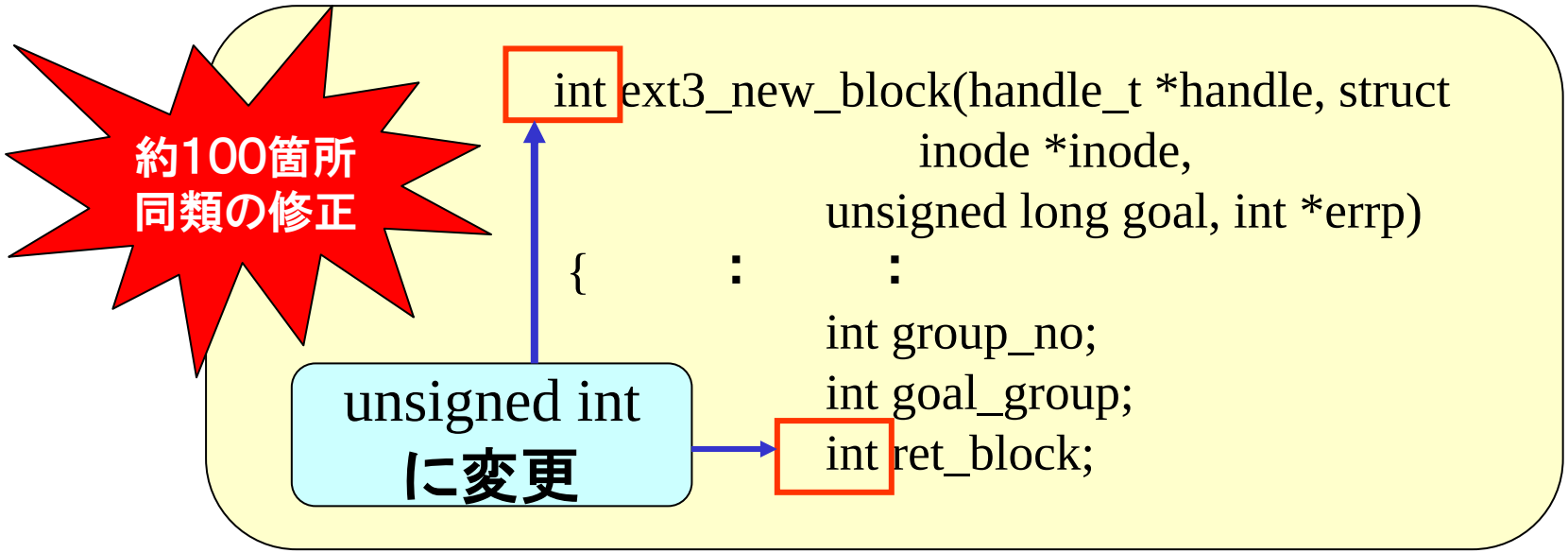
5-2-(4) パッチBの動向

現在の状況、及び今後の予定

- 現在Andreas Dilger氏の指摘事項を反映し、再度パッチ提案済み。議論継続中。

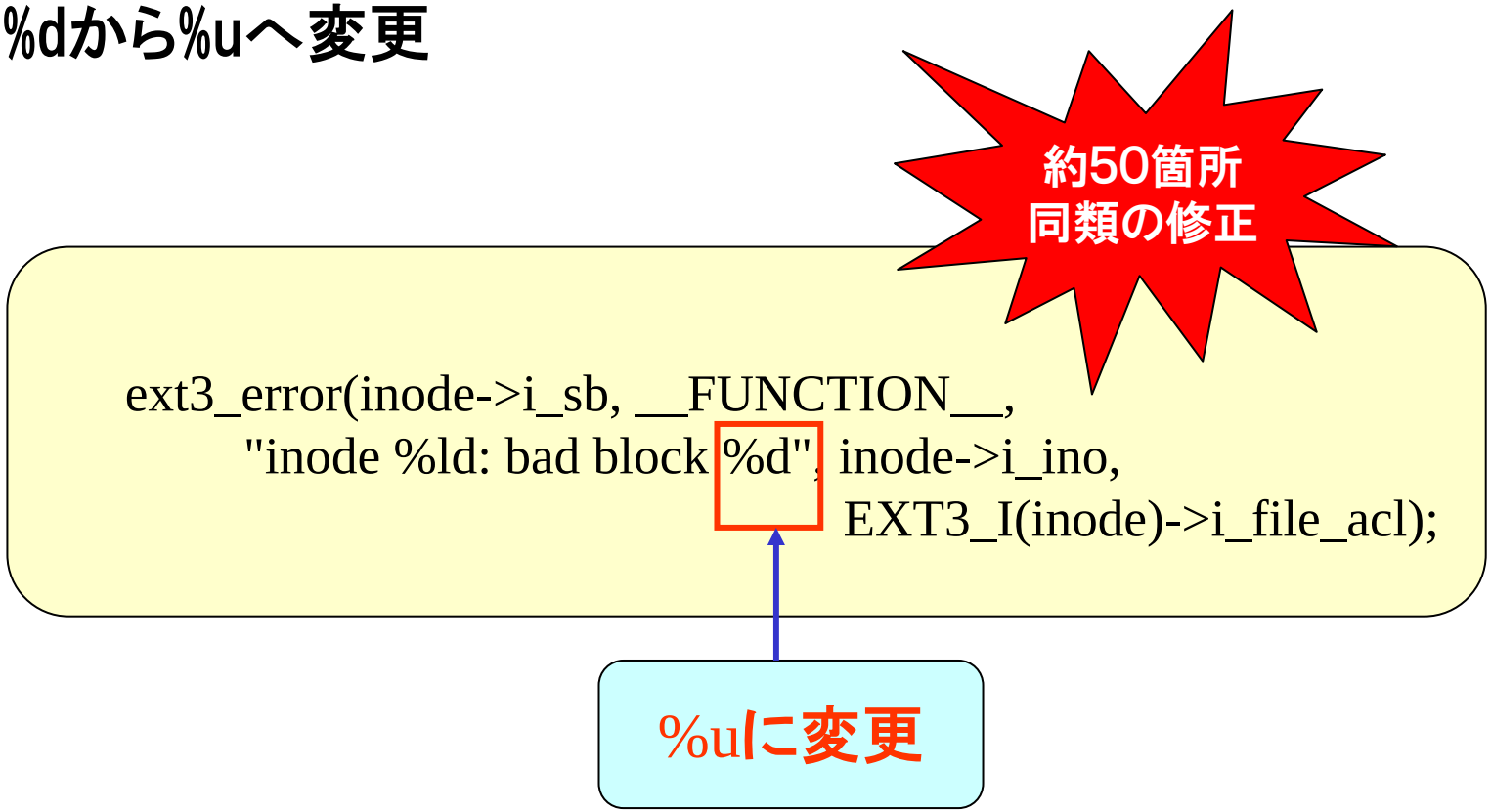
5-3-(1) ファイルシステムサイズの拡大方式

■ ブロックを扱う変数の型を符号付き4バイトから
符号なし4バイトに変更



5-3-(1) ファイルシステムサイズの拡大方式

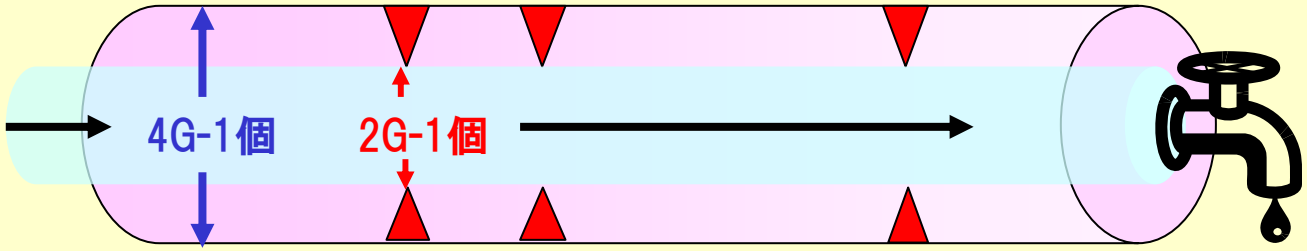
- ブロック番号出力時のフォーマット文字列を %dから%uへ変更



5-3-(1) ファイルシステムサイズの拡大方式

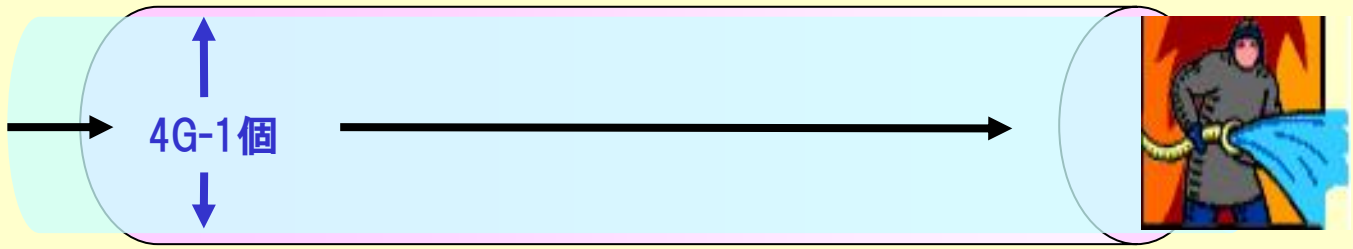
■ ファイルシステムサイズ拡大結果

改造前：（ブロックサイズ）×$(2^{31}-1)$



約2倍に拡大

改造後：（ブロックサイズ）×$(2^{32}-1)$



5-3-(2) コミュニティへの提案、及び議論の動向

- 2006/3/15 ext3開発用メーリングリストに改造パッチを提案
- 2006/4/13 Andreas Dilger氏からの指摘内容

ブロック用変数の型をint, unsigned int等で直接宣言せず、
ブロック種別毎にブロック用の型を定義し使用すべき。

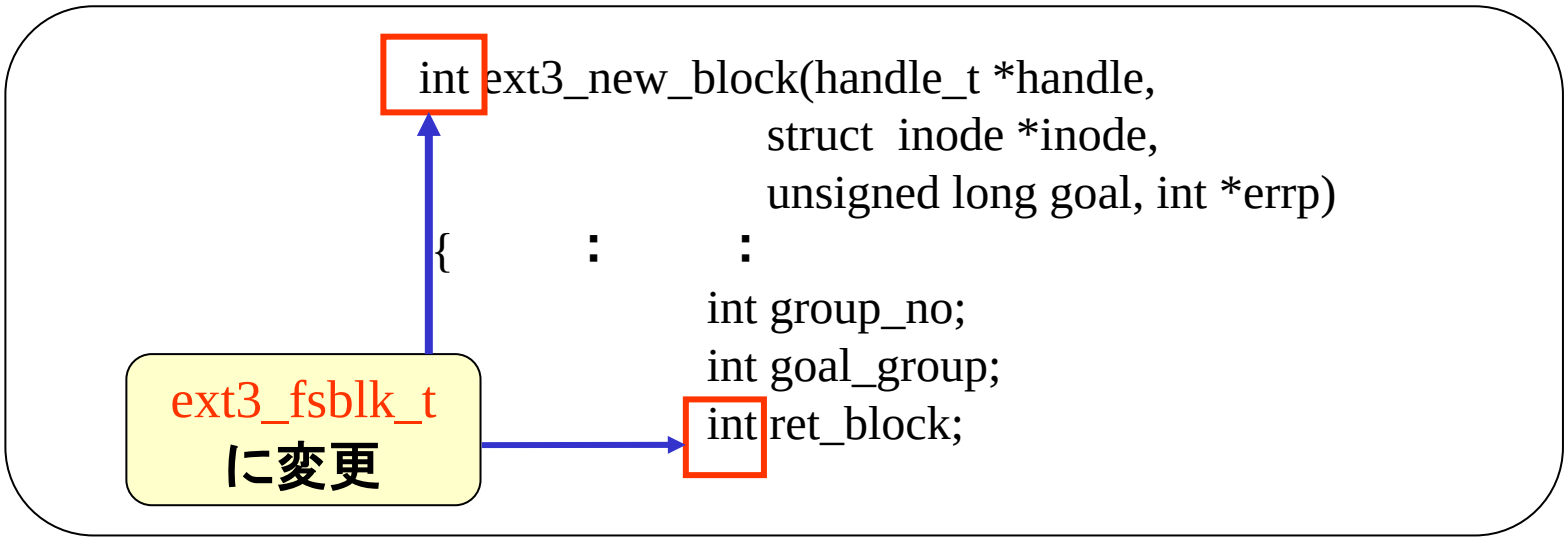
将来的に**少ない修正量でext3の64bit化が可能**となる。

5-3-(2) コミュニティへの提案、及び議論の動向

Andreas Dilger氏からの指摘による修正

ブロック関連変数の型を以下の3つの型に変更

```
typedef unsigned long ext3_fsblk_t; #ファイルシステム相対ブロック番号
typedef unsigned long ext3_fileblk_t;#ファイル相対ブロック番号
typedef long ext3_grpblk_t;          #グループ相対ブロック番号
```



5-3-(2) コミュニティへの提案、及び議論の動向

現在の状況、及び今後の予定

現在Andreas Dilger氏の指摘事項を反映中。
2006/6中旬頃修正パッチを再展開し、議論再開予定。

5-4-(1) ブロックサイズの拡大方式

ext3は元々実装上は最大でページサイズまでの
ブロックサイズを使用可能

マウント時にブロックサイズを最大4KBに制限
しているチェック処理撤廃

最大ブロックサイズ
改造前：4Kバイト

最大16倍

改造後：ページサイズと同じサイズ（最大64Kバイトまで）

※但し、IA64等4KBより大きいページサイズを持つアーキテクチャのみ4KBより
大きいブロックサイズが使用可能となる。

5-4-(2) コミュニティへの提案、及び議論の動向

2006/1/18にext3開発用メーリングリストに改造パッチを提案。

2006/1/18にAndreas Dilger氏より「ディレクトリ先読みブロック数を拡大すべき」との指摘あり。

2006/1/21にAndrew Morton氏より「ディレクトリ先読み関連コードを修正中」とのこと。そのため、我々は修正しないことに決定。

特に反対意見はないため、今後他のパッチと合わせてLinuxへの組込みをメンテナに交渉する予定。

第6章　まとめ

6-1　成果

6-2　今後の課題

NEC

Empowered by Innovation

6-1-(1) 新しい方式による拡大結果

ファイルシステム サイズ拡大
（ブロックサイズ）×($2^{32}-1$)

ファイルサイズ拡大
（ブロックサイズ）×($2^{32}-1$)

ブロックサイズ拡大
アーキテクチャの ページサイズ

ページサイズ	最大FSサイズ		最大ファイルサイズ	
	改造前	改造後	改造前	改造後
4KB	8TB	16TB（約2倍）	2TB	4.1TB（約2倍）
16KB	8TB	64TB	2TB	64TB
64KB	8TB	256TB（約32倍）	2TB	256TB（約128倍）

6-1-(2) 他ファイルシステムとの比較

Linuxの主要ファイルシステムの諸元

FS種別	OS種別	32bit環境		64bit環境	
		最大FSサイズ	最大ファイルサイズ	最大FSサイズ	最大ファイルサイズ
XFS	Linux	16TB	16TB	8EB	8EB
JFS	Linux	16TB	16TB	32PB	4PB
ext3	**Linux**	**16TB**	**4.1TB**	**256TB**	**256TB**
ReiserFS	Linux	16TB	2TB	1EB	16TB

32ビット環境においては他のFSと比較しても遜色ない

6-1-(3) コミュニティとの議論の成果

メモリinodeのi_blocksを8バイトにするパッチがLinuxに取りこまれた

Linuxの全てのファイルシステムで、32bit環境において
2TB以上のファイル作成を可能とする準備が整った。

コミュニティにおいてext3諸元拡大に関する議論の輪が広がっている

我々の提案が発端となり現在コミュニティでは1日に数十通にも
及ぶメールにより、ext3の諸元拡大に関する議論が行われている。

その他のパッチもコミュニティに好意的に受け入れられている

Linuxに取り込まれるようにコミュニティとの議論を継続する。

6-2 今後の課題

ブロック検索の性能向上
ブロック数が増大するためブロック検索性能向上が必要
エクステント等

効率的なブロックアロケーションによるI/O性能向上
ブロック数が増大するため最適なブロック配置を行う機能が必要
遅延アロケーション、デフラグメント 等

ディスク使用効率改善
ブロックサイズ拡大によるディスク使用効率の悪化を防止
テイルパッキング等

さらなる諸元の拡大
ディスクストレージ増大に伴いさらなる諸元拡大が必要
ext3の本格的な64bit化等

NEC

NEC Empowered by Innovation

NEC Software Tohoku, Ltd.