高信頼Fat-Btree構成へのneighbor-WALプロトコルの適用

北陸先端科学技術大学院大学
情報科学研究科
宮崎 純

東京工業大学
学術国際情報センター
横田 治夫

目的

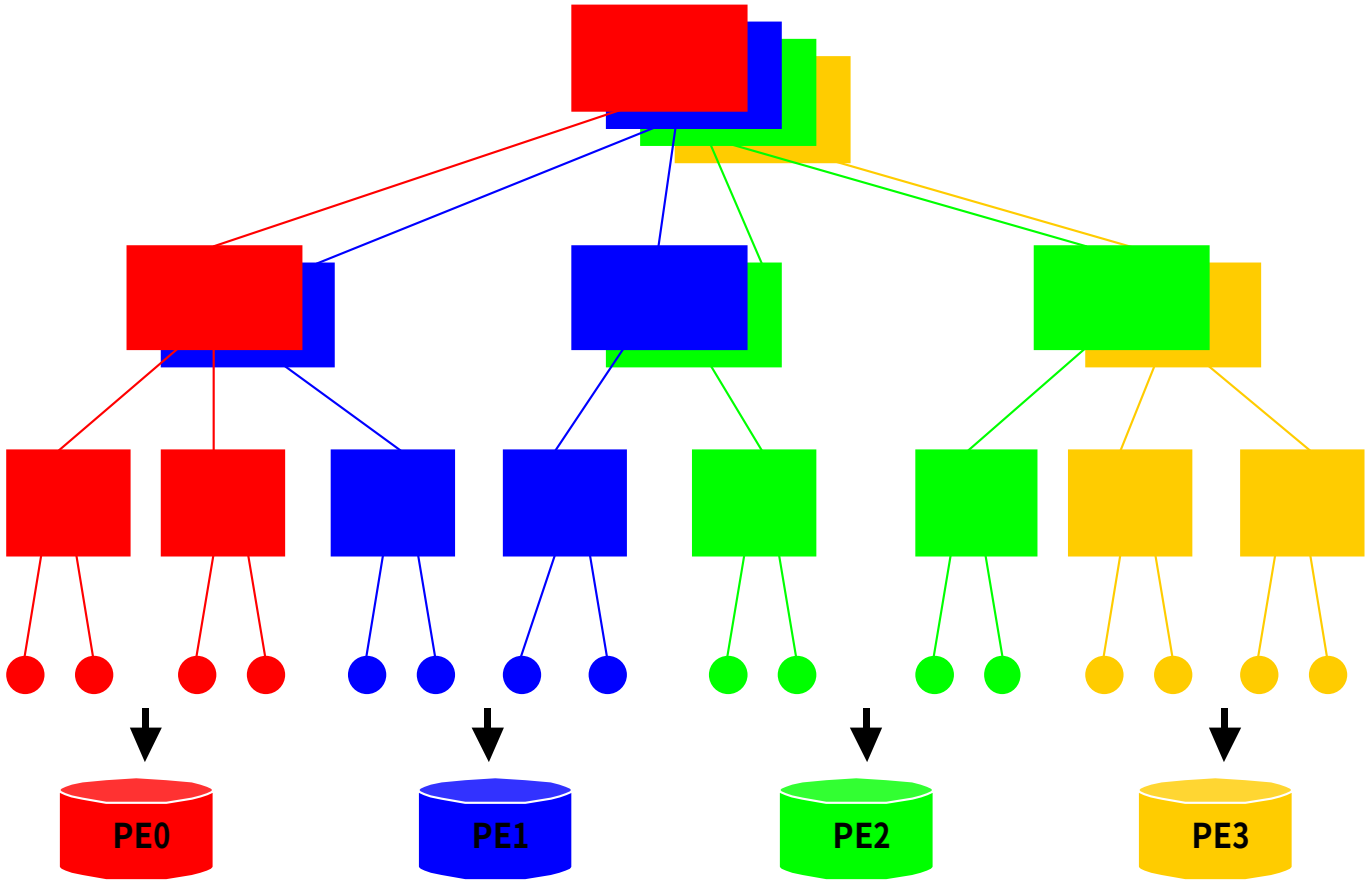
- 高性能並列ディレクトリ構造Fat-Btreeの
  - 高信頼構成方式とその改良
  - 信頼性と性能の比較

あらまし

- Fat-Btree構造
- 高信頼Fat-Btreeの構成方式
- Neighbor-WALプロトコル
- 高信頼Fat-Btreeの性能比較
- まとめと今後の課題

Fat-Btree構造

- 各PEは葉に可達なノードのみを持つ



信頼性の指標

- **平均データ喪失時間**(MTTDL)
  - 1つ以上のデータが完全に失われるまでの平均時間
- **アベイラビリティ**
  - システムが完全サービス可能である確率

    cf. 完全サービス可能＝無故障時のサービスが全て行えること

システムの故障の仮定

- **構成要素の単一故障**
  - ある構成要素 (プロセッサ、メモリ、ディスク等) の故障中あるいはその復旧中に、他の構成要素は故障を起こさない
- **フェイルストップ**
  - あるPEの構成要素が故障を起こせば、そのPEは停止する

障害の分類

- トランザクション障害
  - トランザクションのアボート
- システム障害
  - メモリやプロセッサの故障、プロセスの異常終了により揮発性記憶上のデータが消失
- 媒体障害
  - ディスク等の故障により、不揮発性記憶上のデータが消失

基本的にログを取ることによりリカバリ可能

障害からのリカバリ

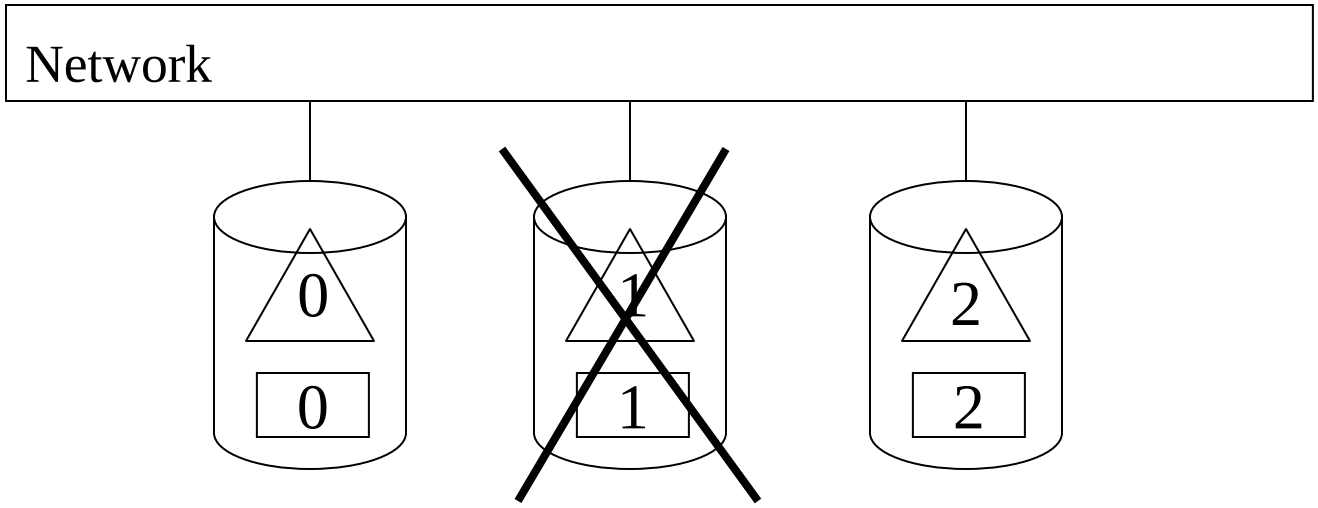
- トランザクション障害
  - ログからリカバリ可能
- システム障害
  - 故障した構成要素を交換後、ログからリカバリ可能
- 媒体障害
  - アーカイブログからリカバリ可能でない場合もある
  - 不揮発性記憶装置（ディスク）の二重化によりリカバリ可能

ログ方式とその性質

- 物理ログ
  - 記憶ブロック（ディスクページ）のバイナリイメージの差分を記録
    - リカバリが高速、ログ量大
- 論理ログ
  - どのデータがどのように更新されたかを論理的に記録
    - リカバリが低速、ログ量小
- 物理論理ログ
  - 記憶ブロック外には物理的、ブロック内は論理的
    - リカバリ、ログ量とも中庸

PL方式

- 各PEごとに物理論理ログを持つ
  - 1PEの媒体障害でデータ喪失

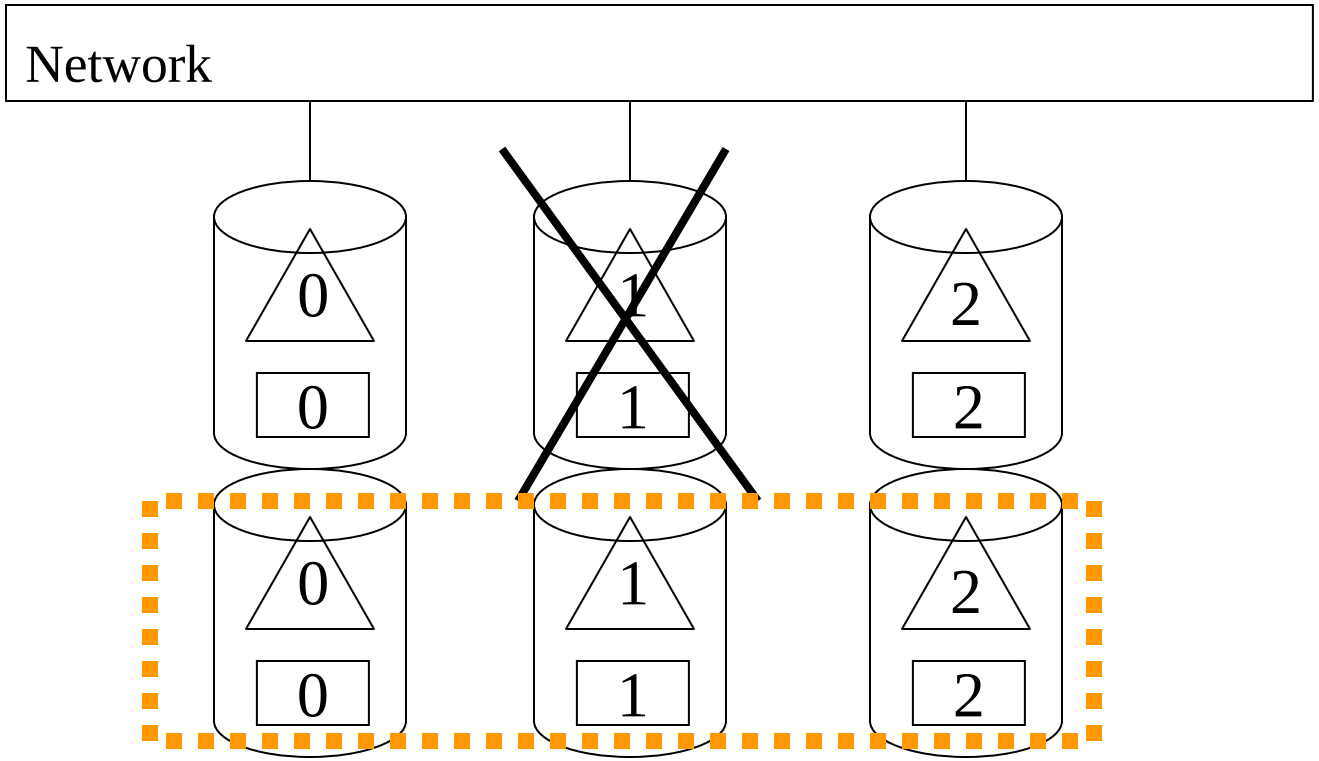


: Fat-Btreeの部分木

: 物理論理ログ

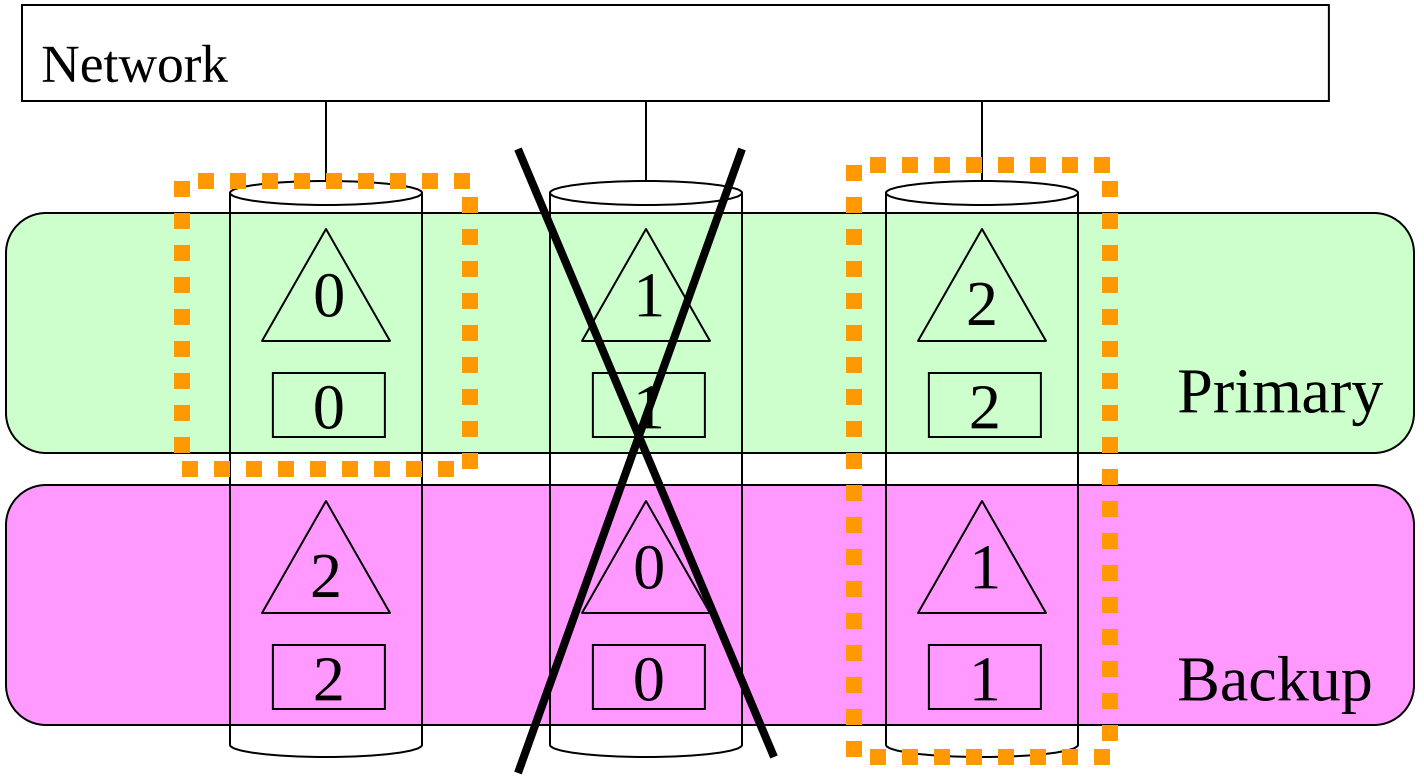
PL+mirror方式

- 各PEでディスクのミラーリング
  - 1PEの媒体障害でも完全サービス可かつ完全回復可
  - ハードウェアコスト高



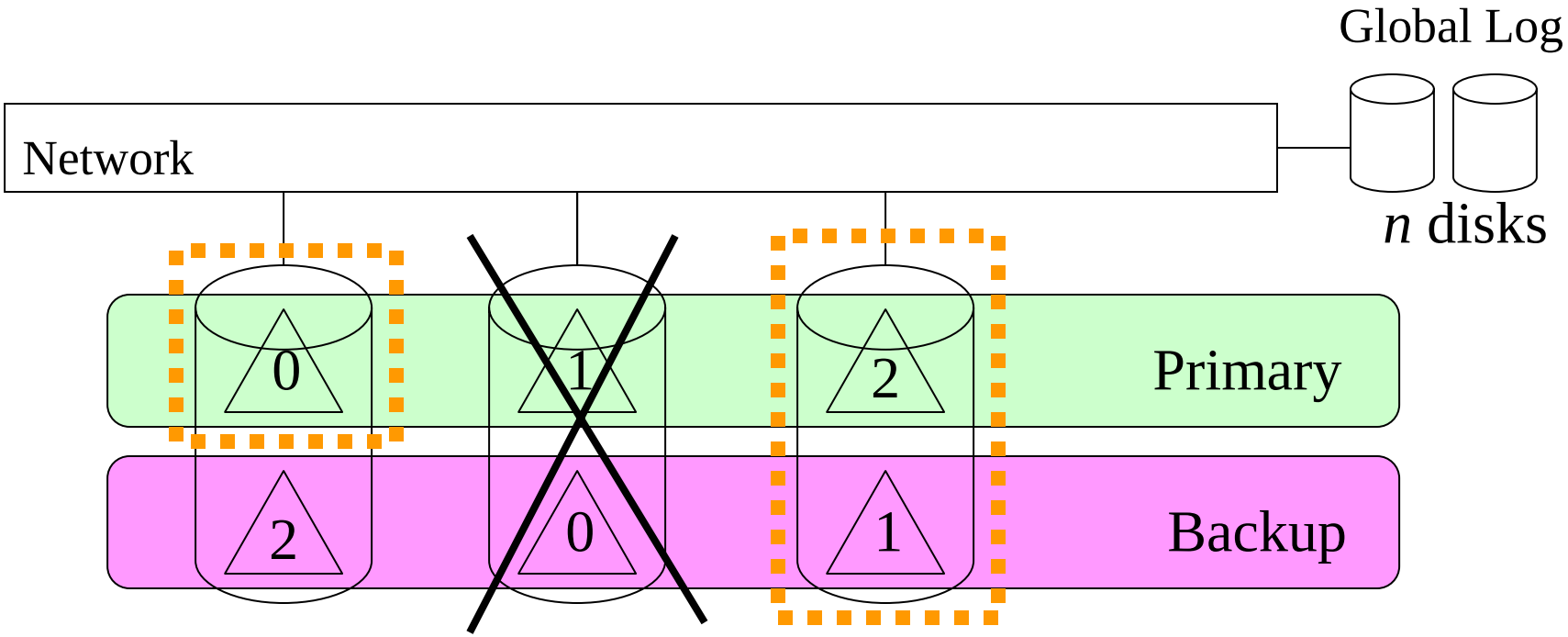
PL+stg方式

- スタガード配置により1disk/PEでも単一故障に対応
  - ハードウェアコスト低
  - プライマリとバックアップの同期更新



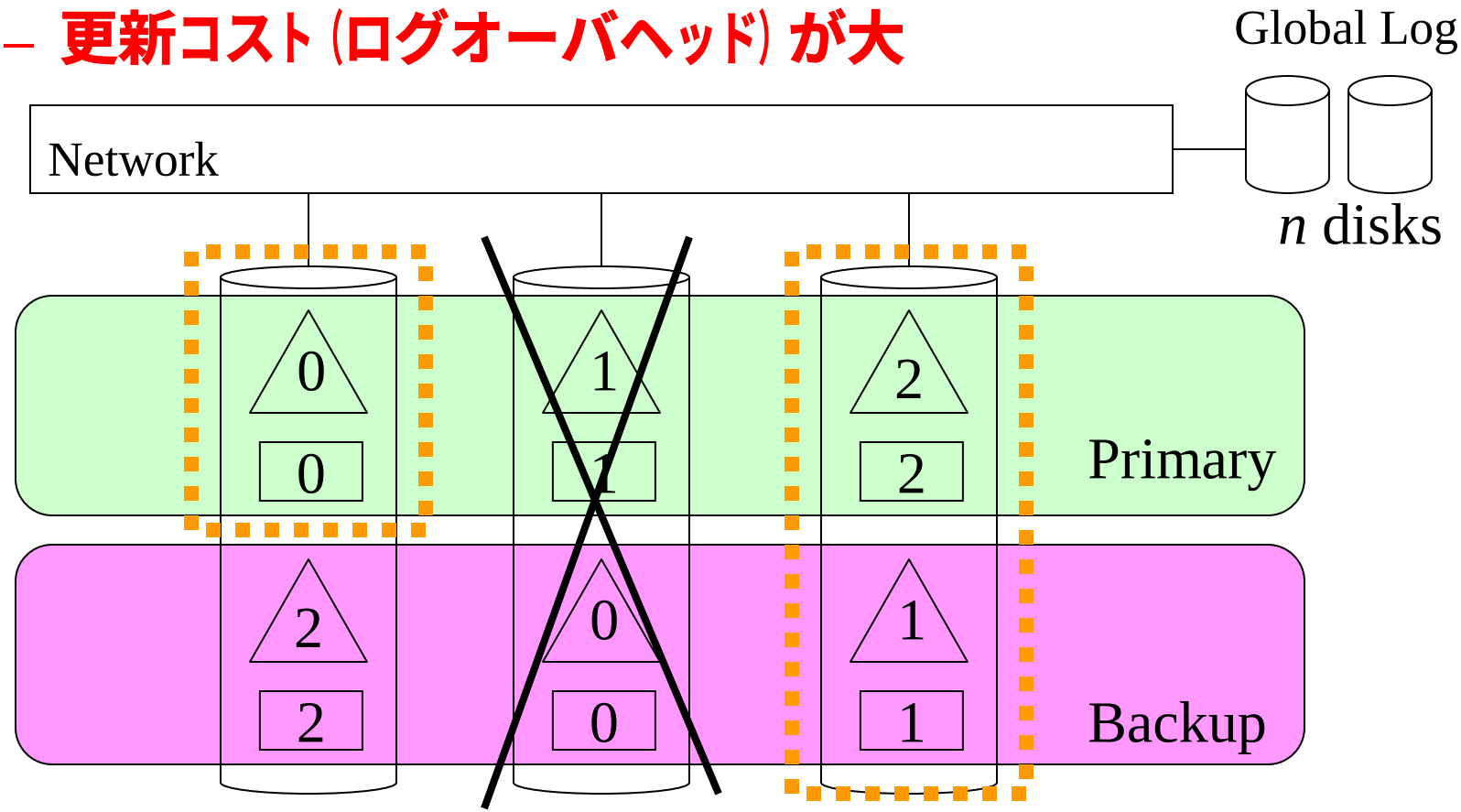
n-GLL+stg方式

- 物理論理ログを無くし、グローバル論理ログを導入
  - 論理ログによりログオーバヘッド小
  - プライマリとバックアップの非同期更新
  - システム障害でもフェイルオーバ(低アベイラビリティ)



n-GLL+PL+stg方式

- *n*-GLL+stgに物理論理ログを導入
  - システム障害でも各PEが独自に回復（高アベイラビリティ）
  - 更新コスト（ログオーバヘッド）が大



高信頼Fat-Btree構成方式の比較

方式	ハードコスト	媒体障害時のサービス	回復可能性	MTTDL	アベイラビリティ	ログオーバヘッド
PL	Low	縮退	データ喪失	Bad	N/A	Low+
PL+mirror	High	完全	完全	Best	High	Low+
PL+stg	Low	完全	完全	Better	High	Med
2-GLL+stg	Low+	完全	完全	Good	Low	Low
1-GLL+PL+stg	Low+	完全	完全	Better	High	High
2-GLL+PL+stg	Low+	完全	完全	Better	High	High

どの方式が良いのか？

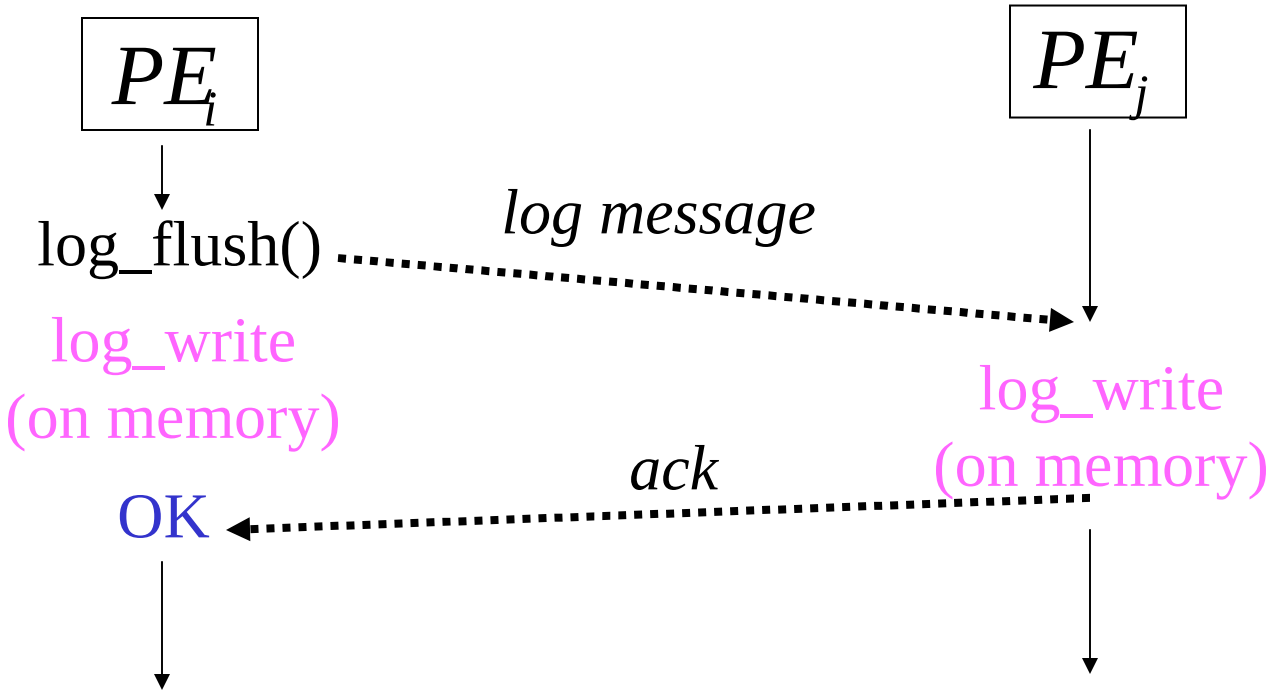
- PL+mirror：**ハードウェアコストが大**
- PL+stg：**プライマリとバックアップの同期更新のため、更新オーバヘッド大**
- 2-GLL+stg：**信頼性が低い**
- 2-GLL+PL+stg：**ログオーバヘッド大**

↓

2-GLL+PL+stg**のログオーバヘッドを下げる**

Neighbor Write-Ahead Log (nWAL)

- **複数のPEの主記憶にログを書いた時点でログが安定とみなす**
  - HypRa(**高アベイラビリティ並列DB**)**で提案**
  - **メッセージ交換はディスクI/Oよりも2,3桁高速**
  - **主記憶オーバフローの時はシーケンシャルにディスクへ**



nWAL適用の効果

- **システム全体の更新スループットの比較**
  - [ ]**内は非同期更新の最悪時の係数**
  - N: PE**数**　Lp: **物理論理ログのコスト**　Ln: nWAL**のコスト**
    U:**更新コスト**　T1: **1**PE**時の最大更新トランザクション数**

	nWAL**適用前**	nWAL**適用後**
PL+stg	$T_{system} = \frac{1}{2} NT_1$	$T_{system} = \frac{1}{2}(1+\frac{L_p - L_n}{U + L_n})NT_1$
2-GLL+stg	$T_{system} = \left[\frac{1}{2}\right](1+\frac{L_p}{U})NT_1$	$T_{system} = \left[\frac{1}{2}\right](1+\frac{L_p}{U})NT_1$
2-GLL+PL+stg	$T_{system} = \left[\frac{1}{2}\right]NT_1$	$T_{system} = \left[\frac{1}{2}\right](1+\frac{L_p - L_n}{U + L_n})NT_1$

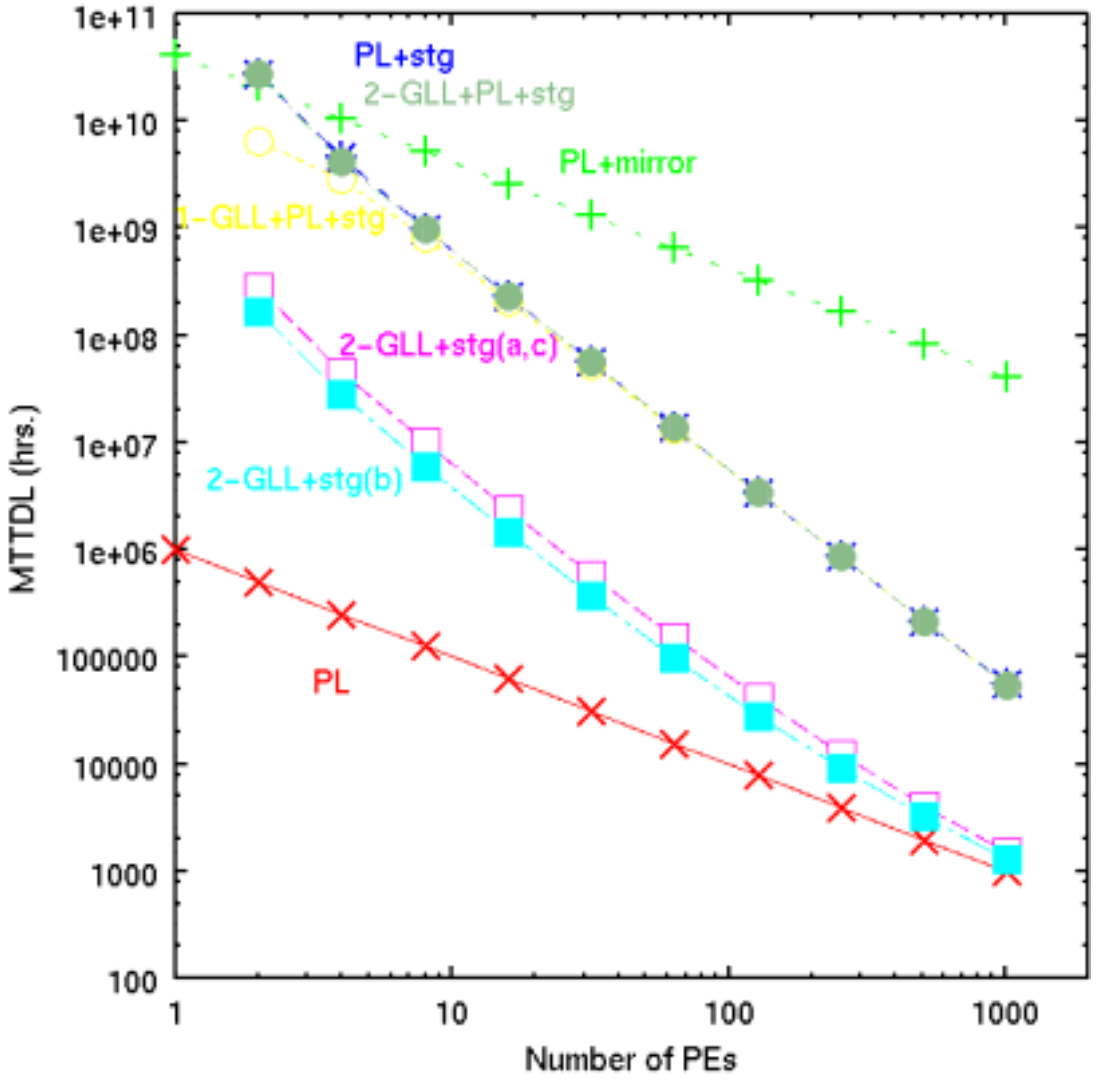
まとめ

- 高信頼･高性能なFat-Btree構成方式
  - コスト、信頼性および性能のバランスは難しい
    - ハードウェアコストを無視： PL+mirrorが最良
    - 低コストかつ高信頼とするためにはログオーバヘッドが大きくなる
  - nWALによるログオーバヘッドの削減
    - *n*-GLL+PL+stgはハイコストパフォーマンス

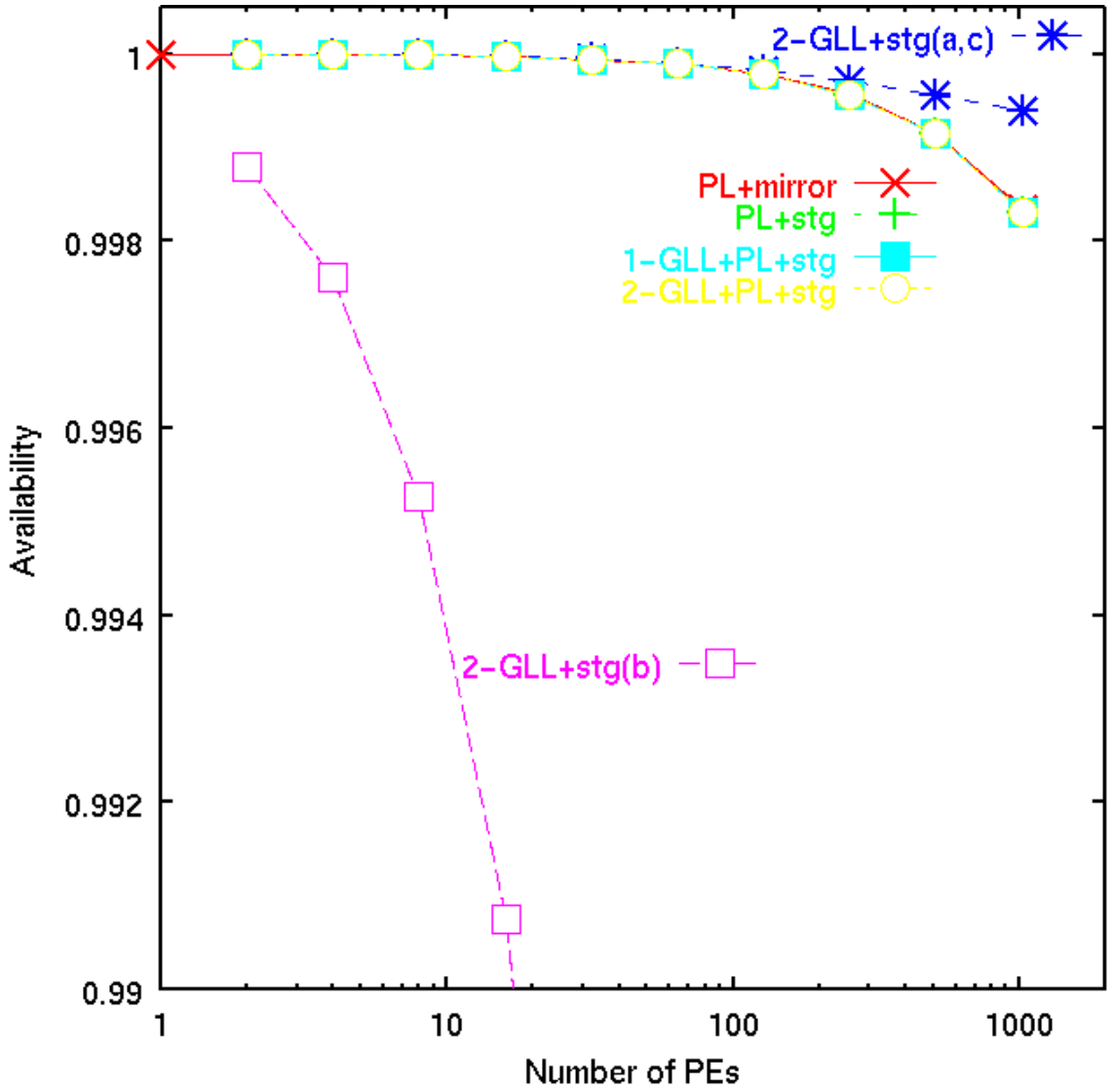
今後の課題

- 高信頼Fat-Btreeの実装またはシミュレーションによる具体的な評価

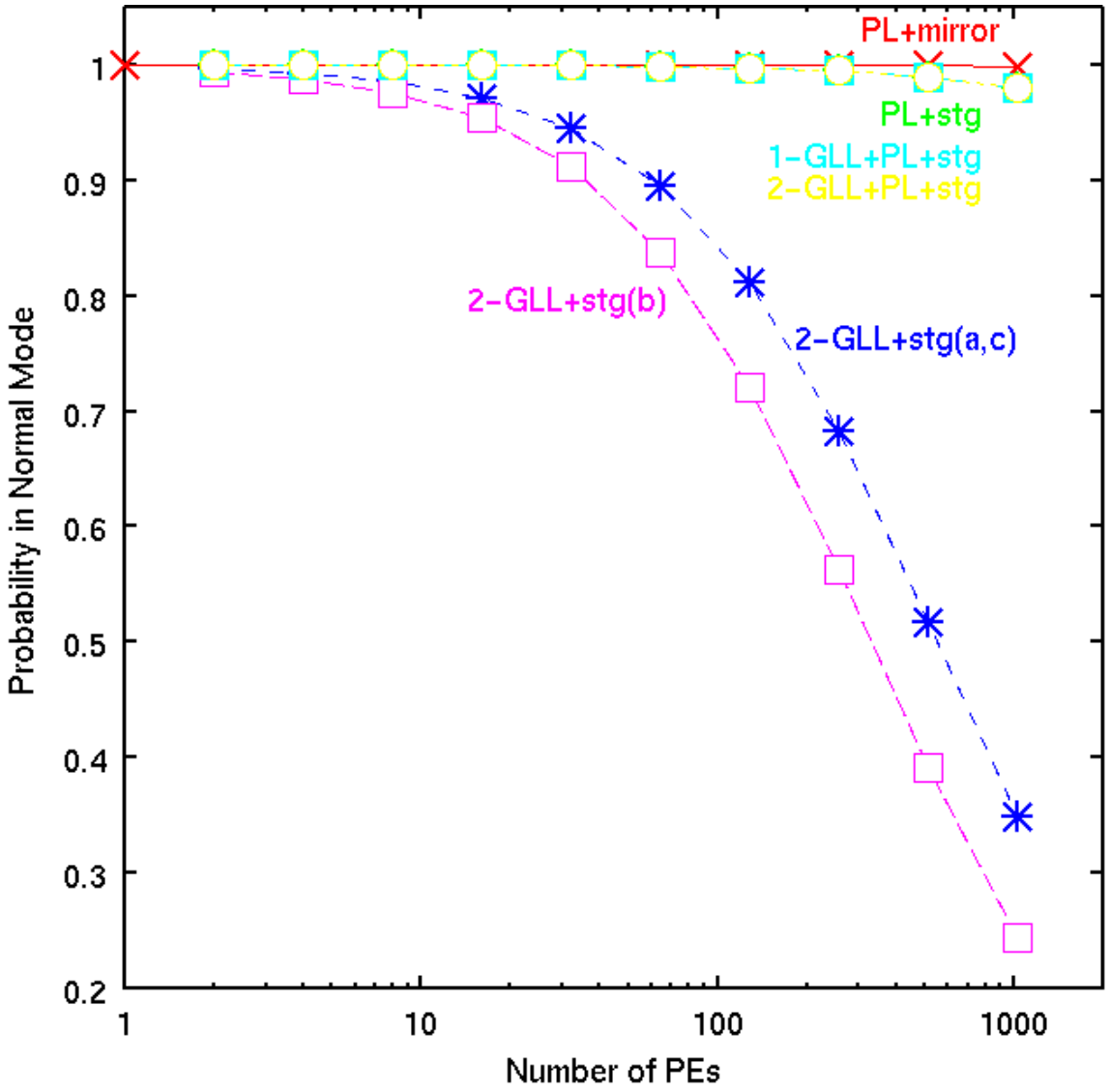
平均データ喪失時間(MTTDL)



アベイラビリティ


2-GLL+stg(a,c)
PL+mirror
PL+stg
1-GLL+PL+stg
2-GLL+PL+stg
2-GLL+stg(b)
Availability
Number of PEs
1
0.998
0.996
0.994
0.992
0.99
1
10
100
1000

無障害運転の確率


PL+mirror
PL+stg
1-GLL+PL+stg
2-GLL+PL+stg
2-GLL+stg(b)
2-GLL+stg(a,c)
Probability in Normal Mode
Number of PEs
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
1
10
100
1000