PostgreSQL Updates

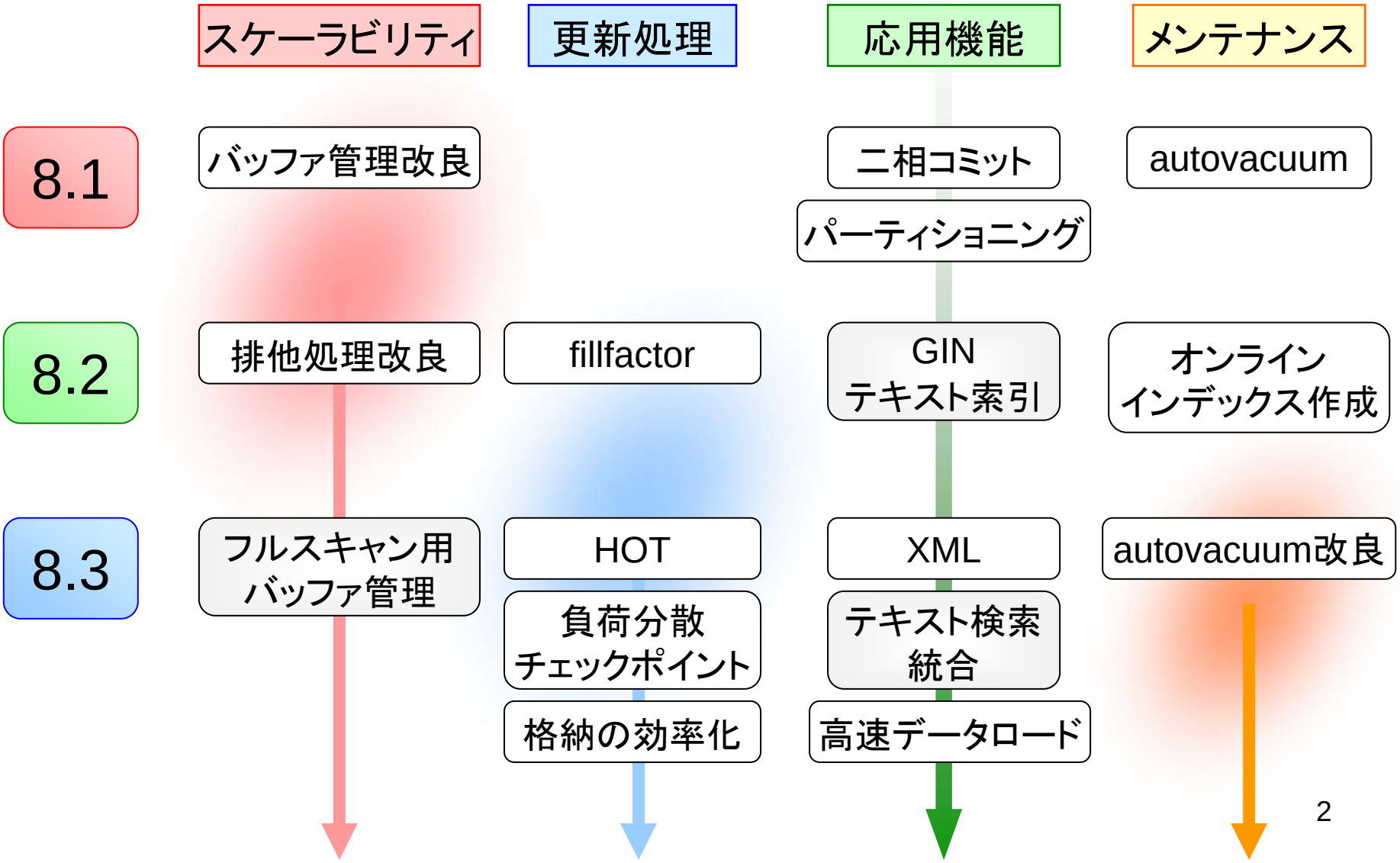
～8.3は更新も速い～

2007.6.23

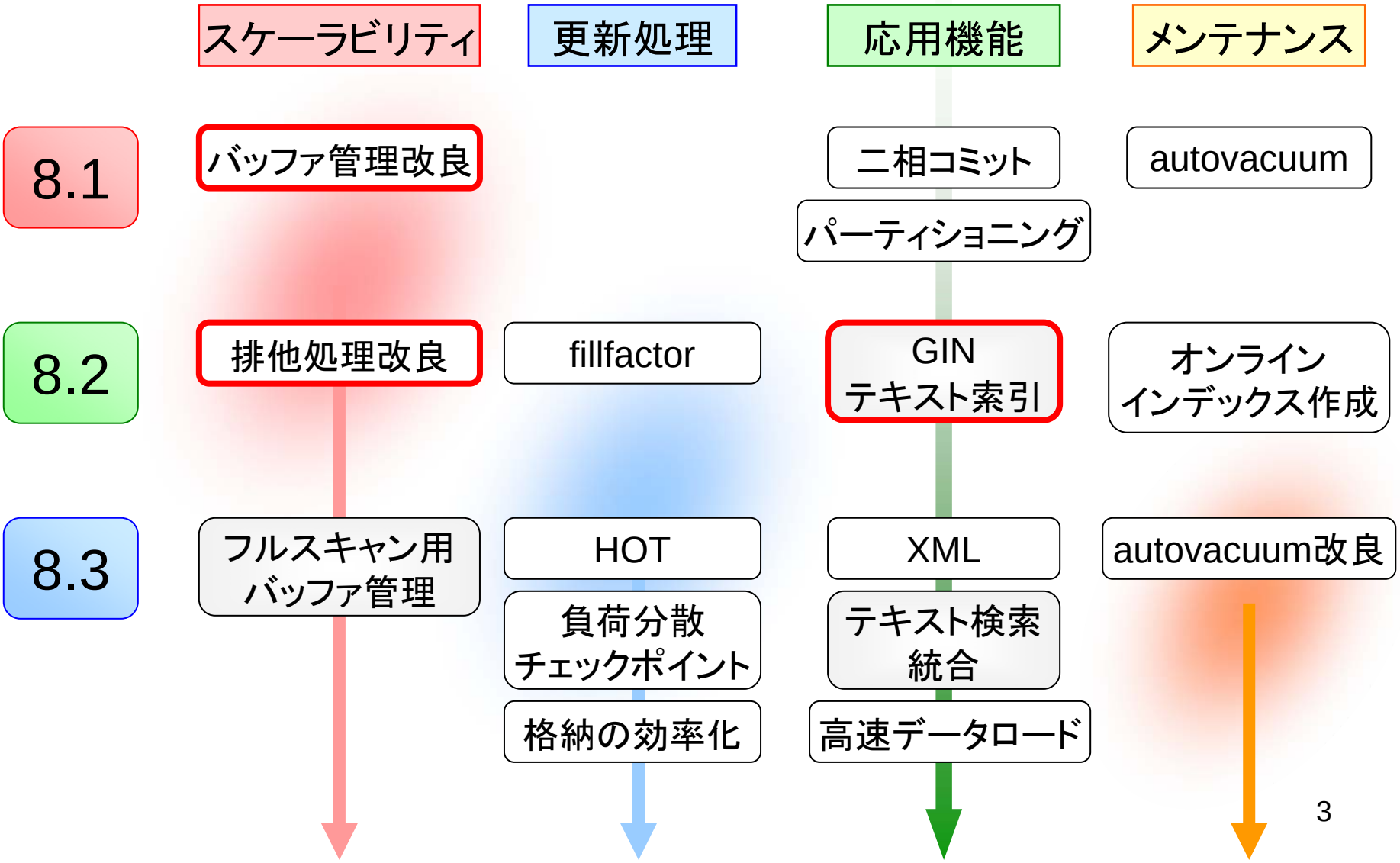
日本PostgreSQLユーザ会

板垣貴裕

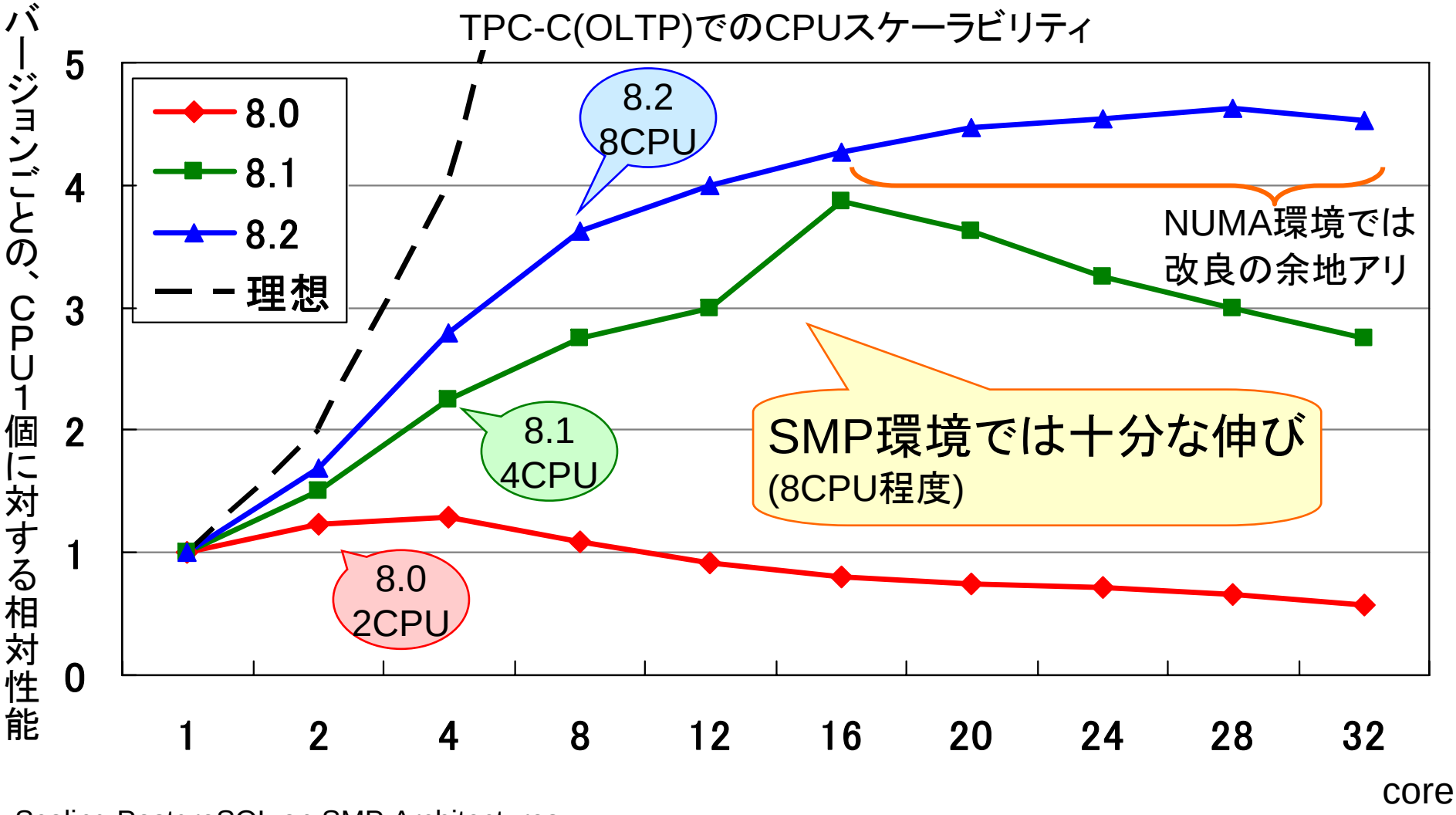
8.1〜8.3 での主要な改善項目



8.1～8.2 の改善項目おさらい



おさらい：CPUスケーラビリティ



Scaling PostgreSQL on SMP Architectures
Doug Tolbert (Unisys), PGCon 2007, Ottawa, 2007-05-24
http://www.pgcon.org/2007/schedule/events/16.en.html を再集計

おさらい：全文テキスト検索

- tsearch2 全文テキスト検索を可能にする拡張

例: **SELECT isbn,title FROM books**
WHERE fts @@ to_tsquery('word1 & word2');

  - GiST, GIN: 特性の異なる2種類のテキスト検索インデックス
  - 共にトランザクション, リカバリをサポート

tsearch2 GiST版 と GIN版 の性能比較

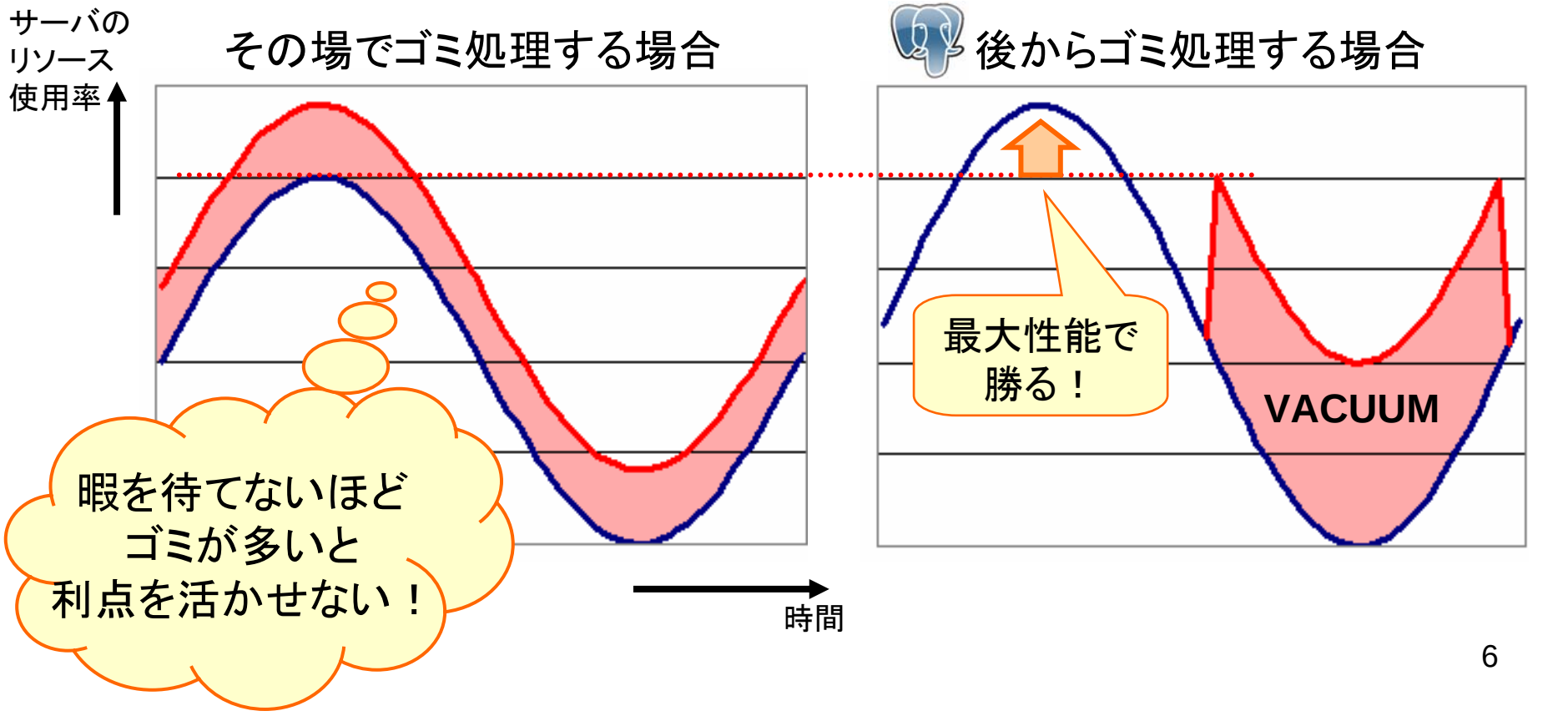
	参照	更新	作成	サイズ
GiST	112ms	280ms	176s	146MB
GIN	39ms	3344ms	532s	306MB
比率	×0.35	×11.94	×3.02	×2.10

使い分けを推奨
•更新が多い → GiST
•参照が多い → GIN

- 日本語を扱うには、ひと手間が必要。まだ「公式」サポートは無い

おさらい：VACUUMによるゴミ処理

- 「追記型」＝UPDATEでゴミが発生する
  - 忙しいときは「ゴミを気にしない」
  - 暇になったら「後からゴミを処理」

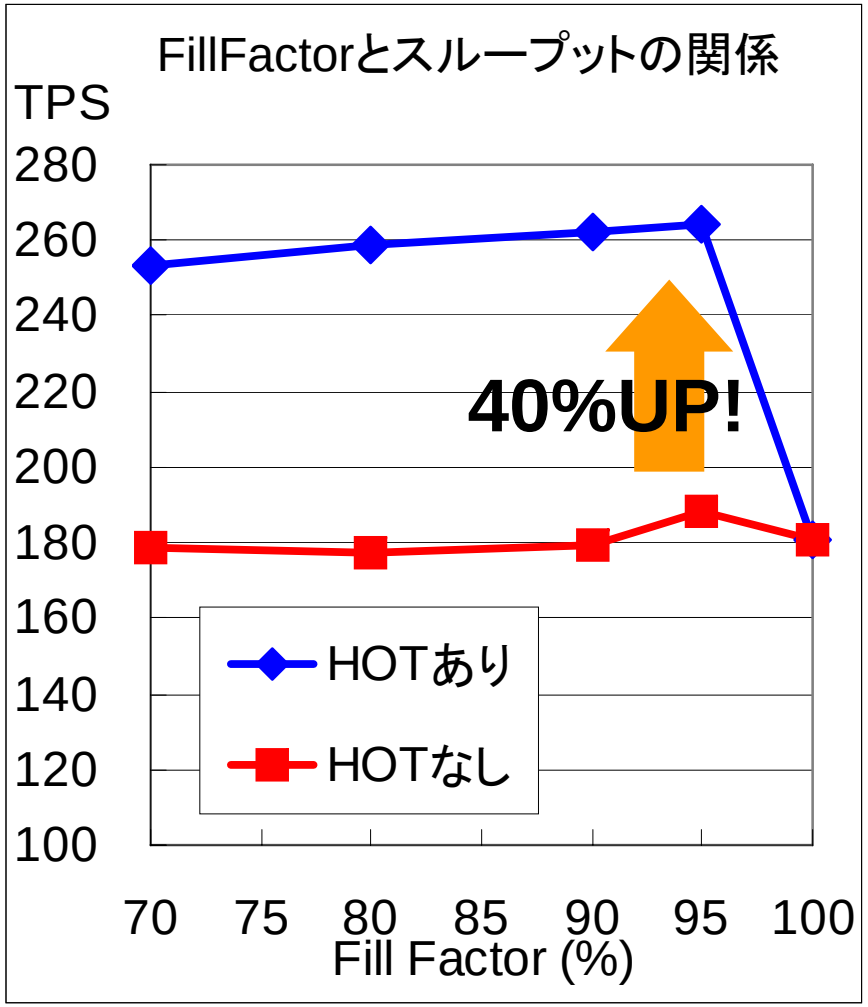


PostgreSQL 8.3 の目玉

1. 更新が速い！
   - HOTでOLTPも得意に
2. 性能が安定！
   - 負荷分散チェックポイント
3. データロードが速い！
   - データロード時にWALをスキップ
4. VACUUMが手間いらず！
   - VACUUMは改良型autovacuumにおまかせ

(1) 8.3は更新が速い！

- 新機能「**HOT**」の効果
  - 更新処理を効率化
  - pgbenchにて
    スループットが40%向上
- 効果
  - インデックスの更新をスキップ
  - VACUUMを待たずにゴミ処理
    - VACUUMの回数を減らせる
- ポイント：FillFactor
  - 初期状態でページの隙間を残す
  - 100%(デフォルト)よりも
    90~95%ほどが良い
    - ALTER TABLE *name*
      SET (fillfactor=95);

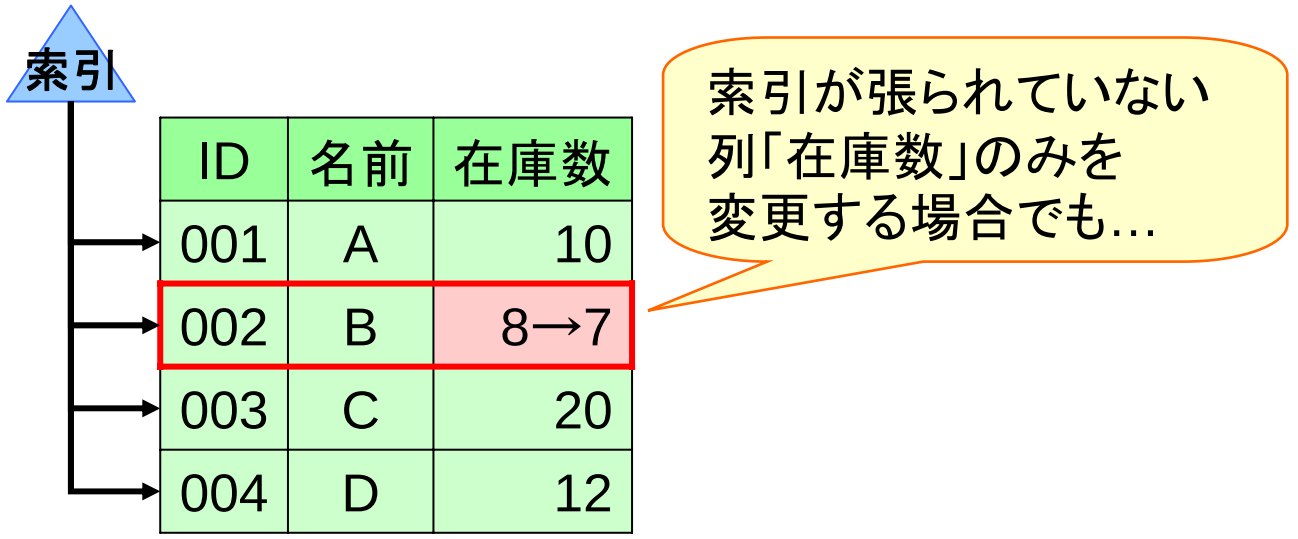


pgbench -s400 (5GB)

NTT OSS Center 調べ

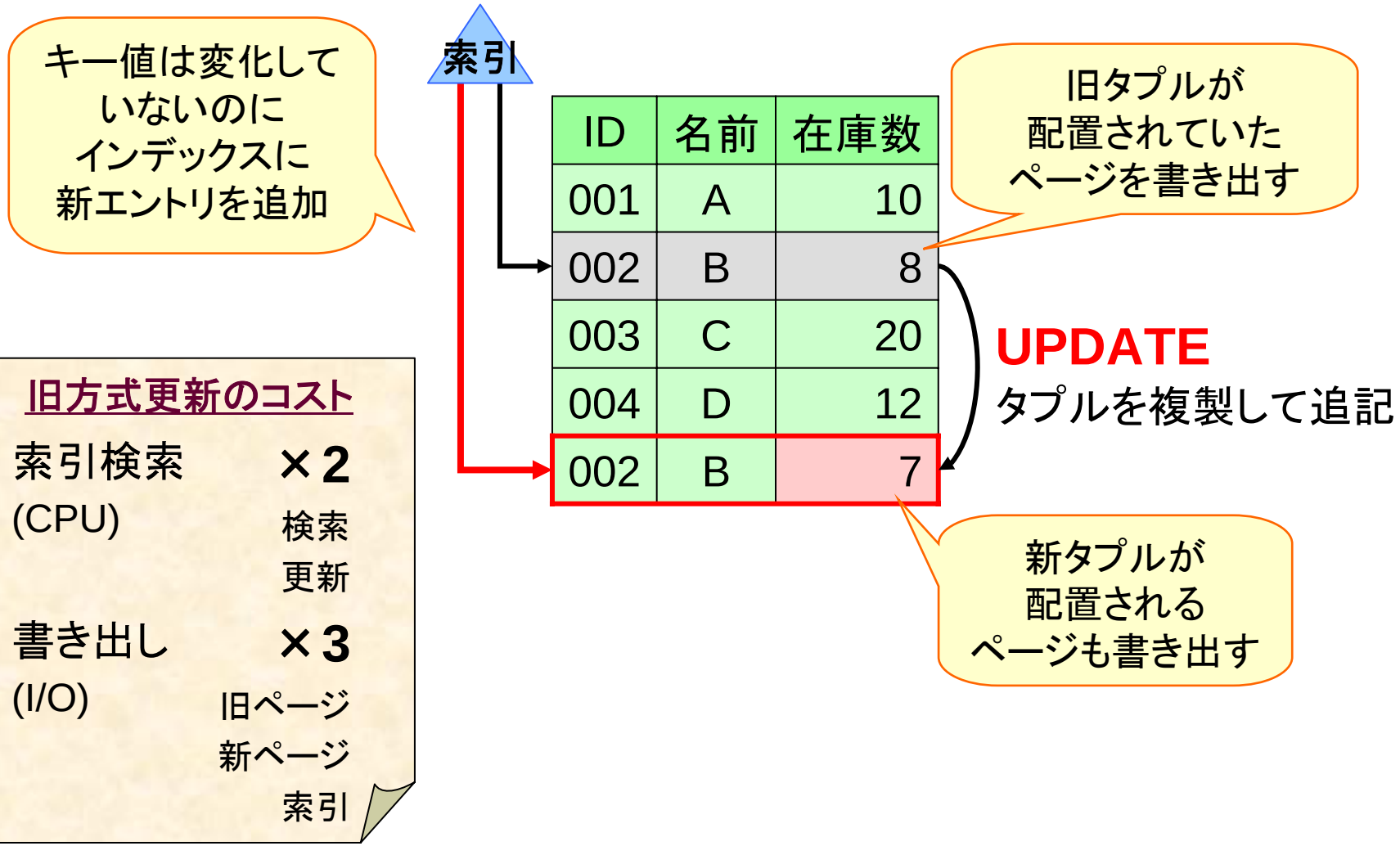
HOT インターナル (1)

- 8.2までのUPDATE時の動作



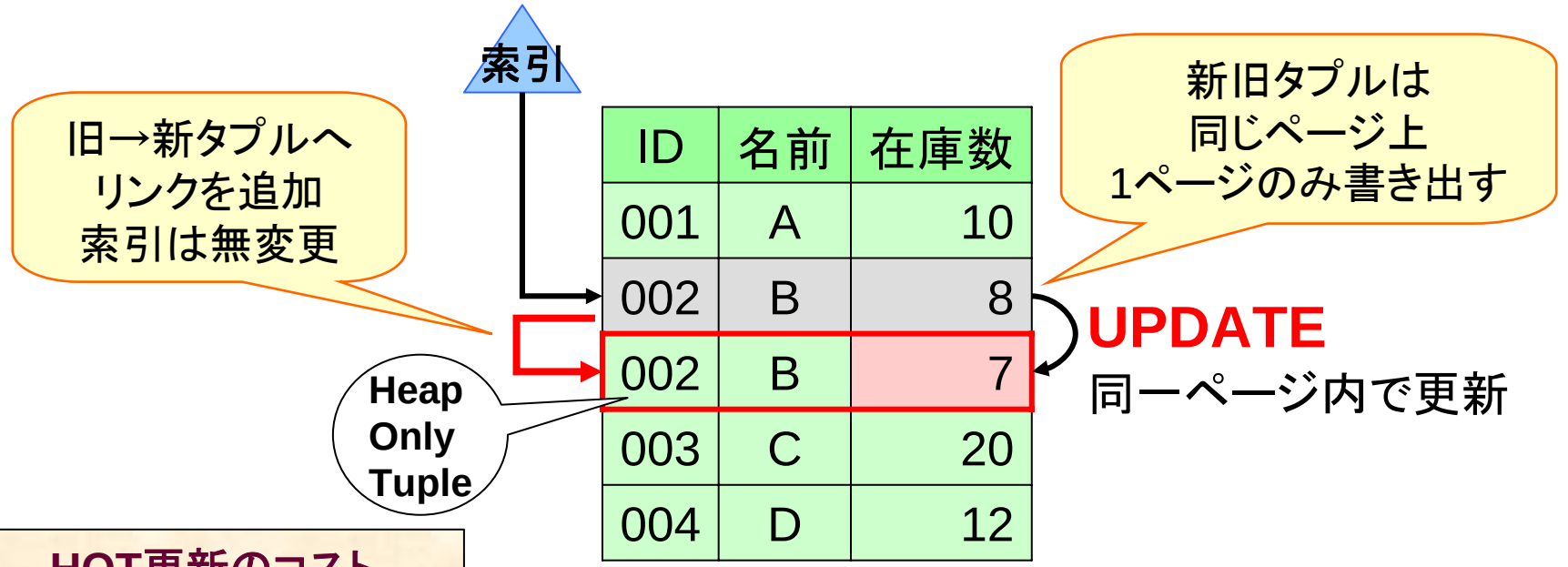
HOT インターナル (2)

- 8.2までのUPDATE時の動作



HOT インターナル (3)

- 8.3 HOT UPDATE時での動作



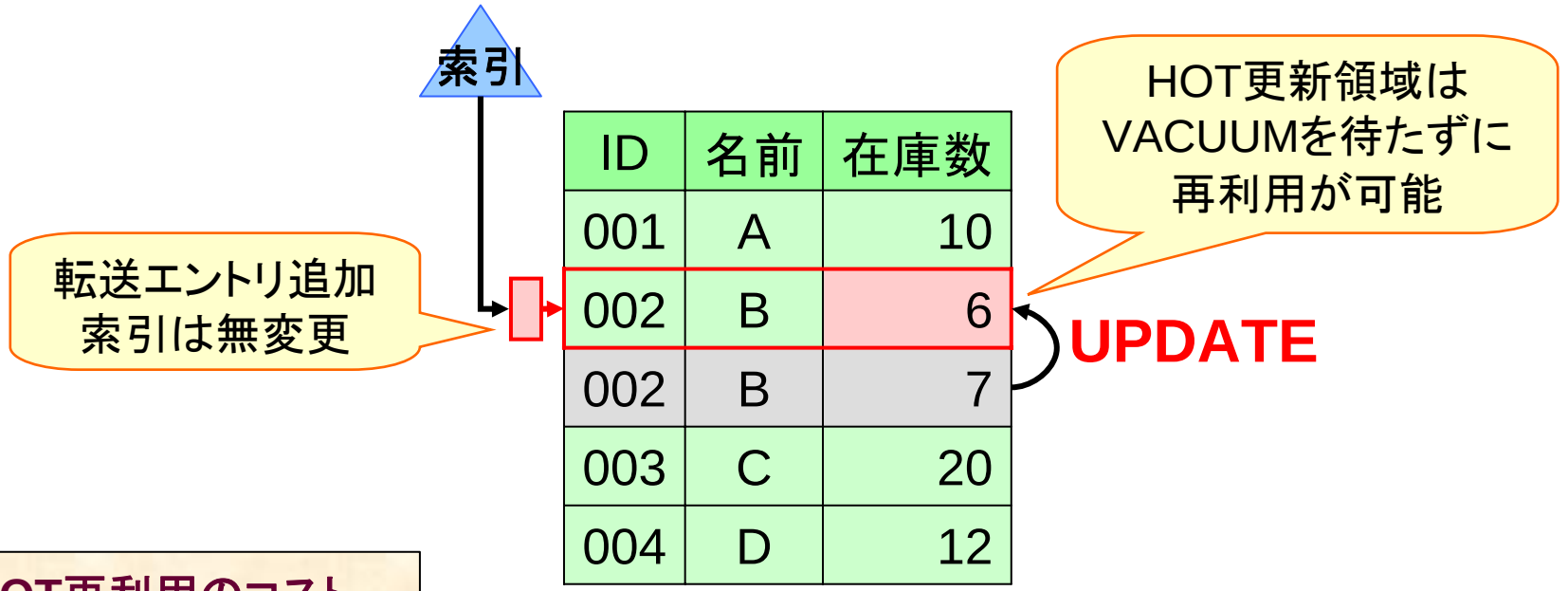
HOT更新のコスト

索引検索 (CPU)	×1 検索
書き出し (I/O)	×1 新旧ページ

計算コスト約1/2に！
書き出しが最大で1/3に！

HOT インターナル (4)

- 8.3 HOT UPDATEの後、不要領域の再利用



HOT再利用のコスト

索引検索 (CPU)	×1 検索
書き出し (I/O)	×1 新旧ページ

単純な更新ならば
VACUUMしなくても
性能を維持可能

HOT更新が利用可能な条件

- UPDATEである
  - DELETE+INSERTはダメ
- インデックスが張られた列を更新しない
  - 「単純な更新」限定だが、この使い方は多い
  - 使われていないインデックスは削除する
- 同一ページ上に空き領域がある
  - FillFactorを調整するとベスト
  - 一度VACUUMをかけるだけでもよい

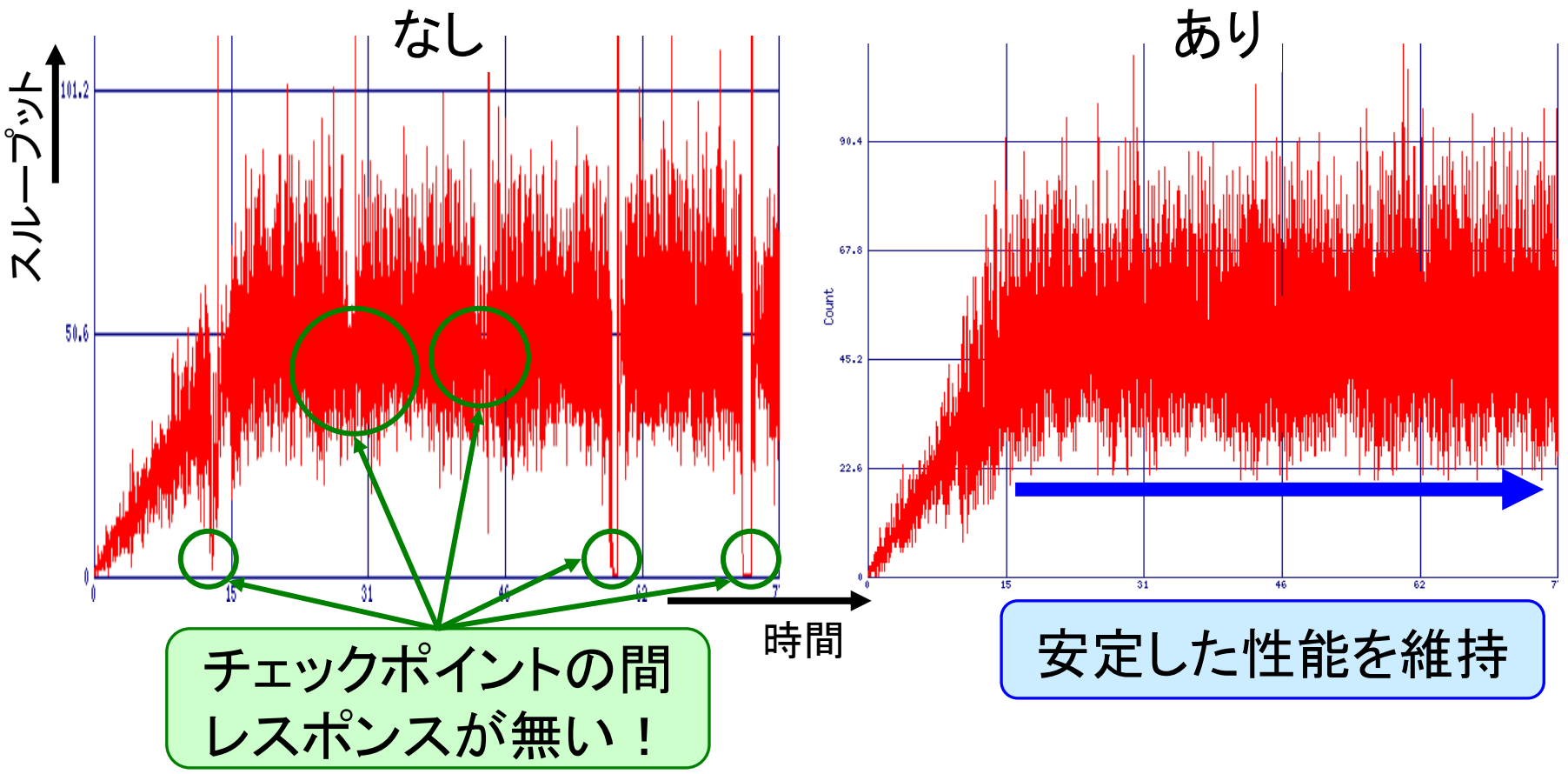
計算コスト、書き出しコストを節約
VACUUMの必要性も削減

PostgreSQL 8.3 の目玉

1. 更新が速い！
   - HOTでOLTPも得意に
2. 性能が安定！
   - 負荷分散チェックポイント
3. データロードが速い！
   - データロード時にWALをスキップ
4. VACUUMが手間いらず！
   - VACUUMは改良型autovacuumにおまかせ

(2) 8.3は性能が安定！

- 負荷分散チェックポイント (LDC)



負荷分散チェックポイント インターナル

- チェックポイント時の動作
  - 全ての変更されたページを書き出す (write)
  - 全ての変更されたファイルを同期する (fsync)

8.2 上記の処理を全速力で行う

8.3 上記の処理を合間にスリープを挟みながら行う

いたって単純

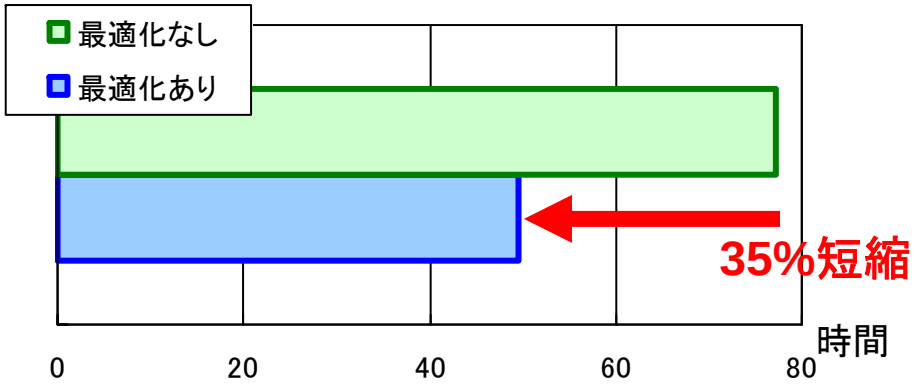
ただし、PostgreSQLに導入された
初の自発的なI/Oスケジューリング
今後のI/O処理の改善に期待

PostgreSQL 8.3 の目玉

1. 更新が速い！
   - HOTでOLTPも得意に
2. 性能が安定！
   - 負荷分散チェックポイント
3. データロードが速い！
   - データロード時にWALをスキップ
4. VACUUMが手間いらず！
   - VACUUMは改良型autovacuumにおまかせ

(3) 8.3はデータロードが速い！

- 新規テーブルへのロードが高速化
  - WALをスキップすることでI/O量が約1/2に



pgbench -i -s30 (テーブルへのCOPY時間のみ)

NTT OSS Center 調べ

高速化するおまじない

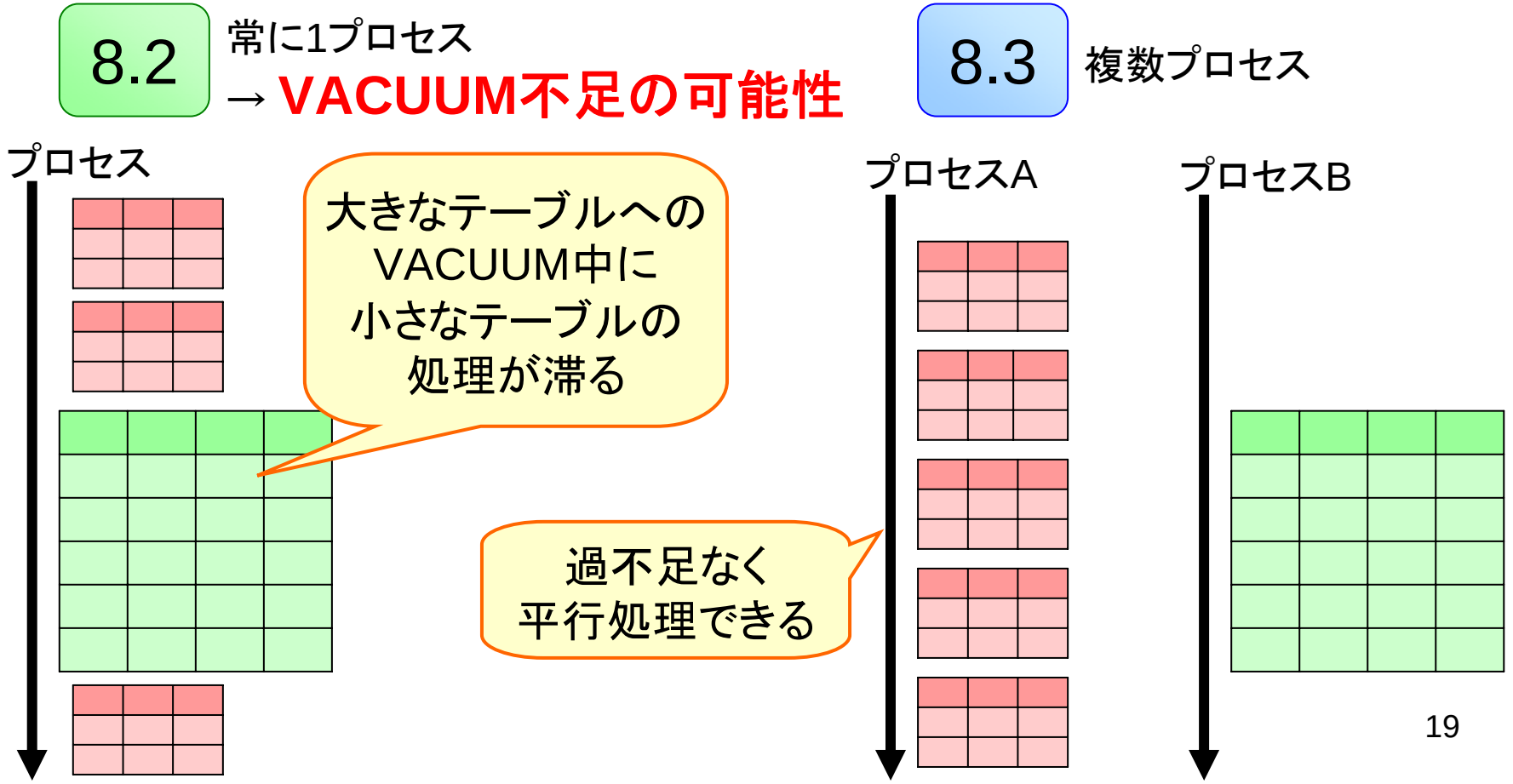
```
BEGIN;
TRUNCATE t;
COPY t FROM …;
COMMIT;
```

- 既存テーブル (インデックス付き) へはまだ苦手
  - 周辺ツールで補う
    - InfoFrame DB Maintenance (NEC)
    - pg_bulkload (オープンソース, pgFoundry)

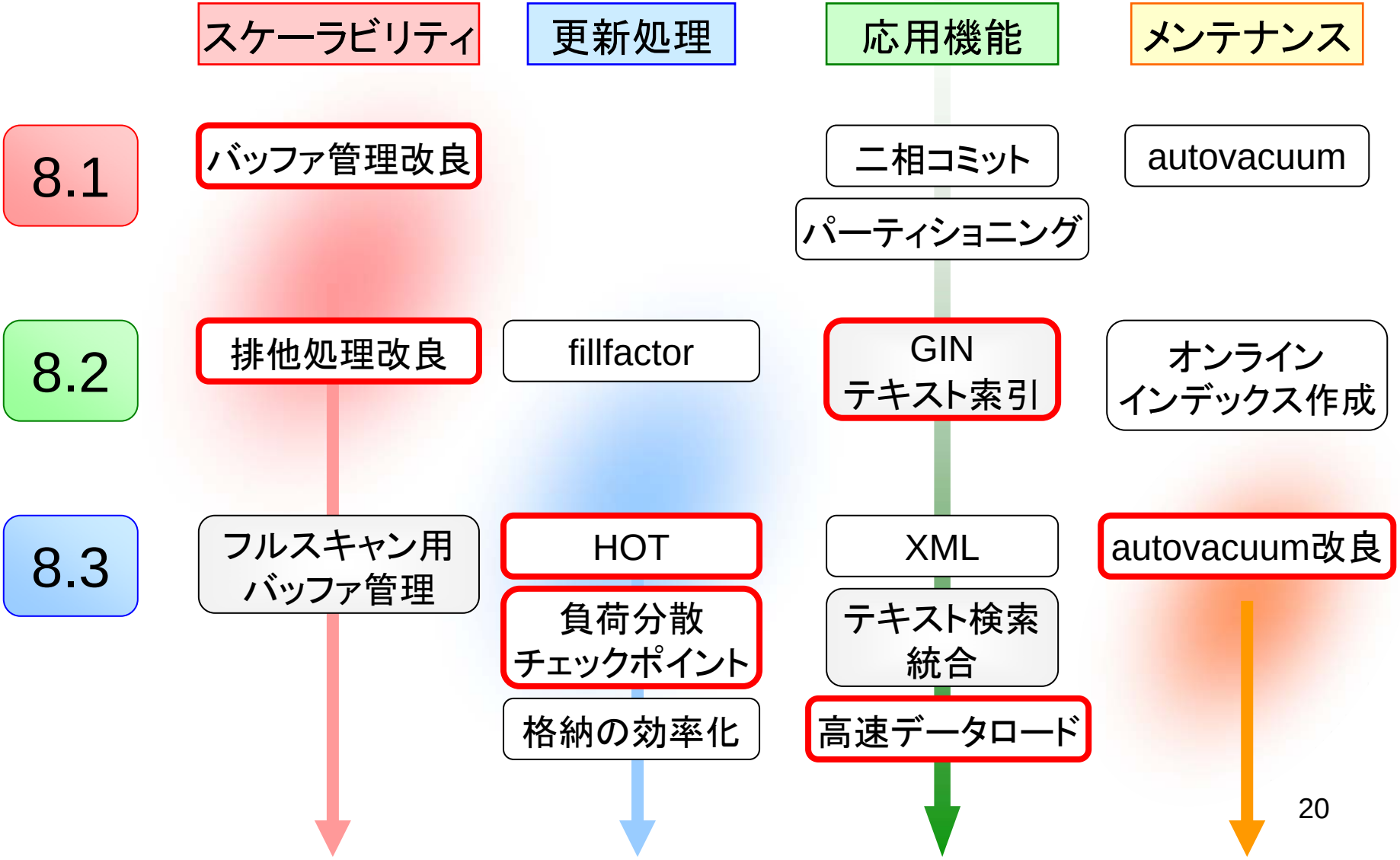
8.3対応よろしく！

(4) 8.3はVACUUMが手間いらず！

- autovacuum によるVACUUMが柔軟になった
  - 複数プロセスを割り当てられる (autovacuum_max_workers)



紹介した主要な改善項目



まとめ

PostgreSQL 8.3 には期待大

- もう、更新が遅いとは言わせない！
  - HOTを初めとする各種OLTP向けの改良
- チューニングと運用の手間が軽減
  - VACUUM, Background Writer (チェックポイント用)
- その他 様々な機能追加や改善
  - SQL/XML, 実行プランアドバイザ, 遅延コミット, etc.

PostgreSQL 8.4 の足音が聞こえる…

- HOTのリミッタ解除
  - 実装をシンプルにするために、現状あえて最適化していない箇所も
- autovacuumスケジューラ
  - メンテナンス時間帯を考慮したVACUUMの実施
- ホットスタンバイ型 負荷分散レプリケーション
  - WALをスタンバイ側へ転送 & スタンバイ側で参照処理を許可

参考：Google Summer of Codes
http://code.google.com/soc/postgres/about.html