

DB2 テクニカル・セミナー 2016

～DB2 for z/OS 最新化がもたらす価値～



DB2 12 for z/OS :
テクニカル・ハイライト および 主要なパフォーマンス・フィーチャー

IBM Silicon Valley Lab.

DB2 for z/OS Development

Akiko Hoshikawa



本日の内容

- DB2 12の主要フィーチャーと継続的デリバリー
- システム・パフォーマンス
- アクセス・パスと照会パフォーマンス
- クラウド、モバイル・アプリケーション
- DBA、365日24時間の可用性

DB2 12の主要フィーチャー



システム・パフォーマンス

- メモリー最適化の活用
- 連続したBPOOL
- INSERTアルゴリズム
- コミットLRSN処理改善
- zIIPサポートの拡大
- スケラビリティの向上
- DRDA LOAD
- 静的SQLのためのRLF



SQL パフォーマンス

- 動的プラン固定化
- RUNSTATS
プロファイル
- UNION ALL、OUTER
JOIN照会でのCPU向上
- SORTの向上
- 適応索引
- UDFキャッシュ



アナリティクス、クラウド、モバイル

- IDAA
- RestAPI
- ページネーション
- 断片ごとの削除
- XM、JSON処理の向上
- 複雑なトリガー
- テンポラル関連拡張
- 拡張MERGE
- SQLPLの向上



DBA、365日24時間の可用性

- 簡易な移行
- オンライン・スキーマ
- PBR RPN
- セキュリティー
- 非同期LOCK二重化
- プロファイル利用の向上
- テンポラルRTS
- ユーティリティの向上
- LOBのzEDC圧縮

DB2 12の主要フィーチャー と 継続的デリバリー

システム・パフォーマンス



- メモリ最適化の活用
- 連続したBPOOL
- INSERTアルゴリズム
- コミットLRSN処理改善
- zIIPサポートの拡大
- スケーラビリティの向上
- DRDA LOAD
- 静的SQLのためのRLF

SQL パフォーマンス



- 動的プラン固定化
- RUNSTATS
プロファイル
- UNION ALL、OUTER JOIN照会でのCPU
向上
- SORTの向上
- 適応索引
- UDFキャッシュ

アナリティクス、クラウド、モバイル



- IDAA
- RestAPI
- ページネーション
- 断片ごとの削除
- XM、JSON処理の向上
- 複雑なトリガー
- テンポラル関連拡張
- 拡張MERGE
- SQLPLの向上

DBA、365日24時間の可用性



- e簡易な移行
- オンライン・スキーマ
- PBR RPN
- セキュリティー
- 非同期LOCK二重化
- プロファイル利用の向上
- テンポラルRTS
- ユーティリティの向上
- LOBのzEDC圧縮



継続的デリバリー(Continuous Delivery)

- 継続的デリバリーのためのインフラストラクチャー更改

継続的デリバリー

コンシューマブルな機能を**スピード**と**品質**とともに市場に提供



継続的デリバリーの基盤



• 検討された案

- リリース・サイクルの早期化 – DB2 vNextを18カ月で
 - DB2 CACのお客様から難色
- 連続したリリース・サイクル
 - DB2 V12.1、V12.2、V12.3、…と続けて、V12の受注ごとにその時点のリリースを提供
 - 移行/共存の課題がかなり大きいほか、リリース進行のコストが高い
- サービス・オンリーと継続的デリバリーのストリームの組み合わせ
 - この案にもDB2 CACから難色

継続的デリバリーの単一ストリーム

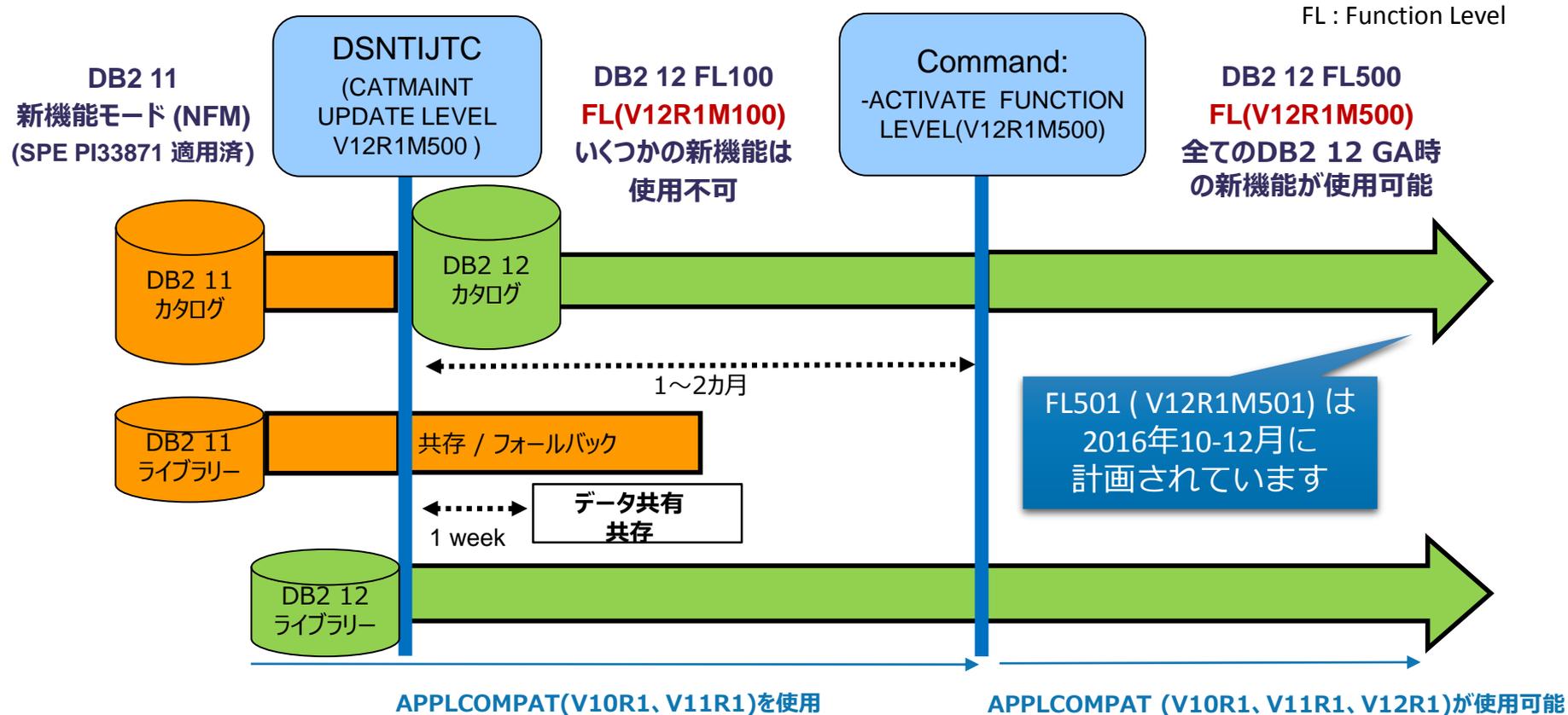
新機能は、事前に決められた日程ではなく、十分な品質になった時点でご提供します

新しくサポートする用語とテクノロジー

- 機能レベル(Function Level; FL)
 - 機能レベルのアクティブ化によって、新機能を取り入れる程度をお客様が管理
 - -ACTIVATE FUNCTION LEVEL、APPLCOMPAT
 - 指定したレベルと、その以前のレベルからの機能がすべてアクティブ化
 - FLアクティブ化はグループ・スコープ – 全メンバーとも、新しいFLで実行できることが条件
 - DB2 12 FLのV12R1M100は共存モード、V12R1M500は最初の新機能レベル
- カタログ・レベル(Catalog Level; CL)
 - カタログの変更は、それを要求するFLをアクティブ化しない限り求められない
- メンテナンス・レベル(Maintenance Level; ML)
 - DB2のメンテナンスや新機能の出荷に伴って、機能レベルとカタログ・レベルは特定のメンテナンス適用が前提条件になる
 - 予防メンテナンスや新機能の追加には、カタログまたは機能レベルの更新は要求されない



移行の流れ DB2 11 NFM -> DB2 12 以降



DB2 12の主要フィーチャー



システム・パフォーマンス

- メモリー最適化の活用
- 連続したBPOOL
- INSERTアルゴリズム
- コミットLRSN処理改善
- zIIPサポートの拡大
- スケーラビリティの向上
- DRDA LOAD
- 静的SQLのためのRLF



SQL パフォーマンス

- 動的プラン固定化
- RUNSTATS
プロファイル
- UNION ALL、OUTER
JOIN照会でのCPU向上
- SORTの向上
- 適応索引
- UDFキャッシュ



アナリティクス、クラウド、モバイル

- IDAA
- RestAPI
- ページネーション
- 断片ごとの削除
- XM、JSON処理の向上
- 複雑なトリガー
- テンポラル関連拡張
- 拡張MERGE
- SQLPLの向上



DBA、365日24時間の可用性

- 簡易な移行
- オンライン・スキーマ
- PBR RPN
- セキュリティー
- 非同期LOCK二重化
- プロファイル利用の向上
- テンポラルRTS
- ユーティリティの向上
- LOBのzEDC圧縮

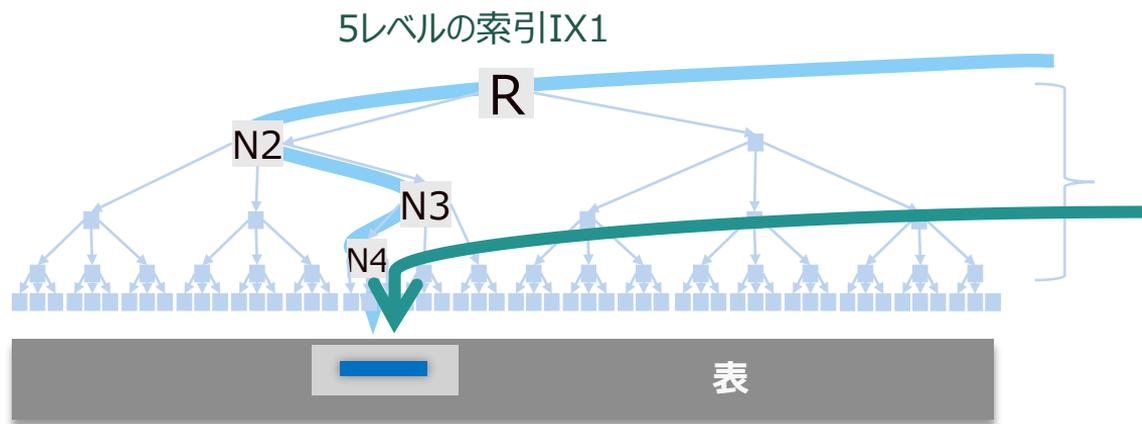
DB2 12の主要フィーチャー – システム・パフォーマンス



システム・パフォーマンス

- 大きな実メモリーの活用
 - メモリー内索引 = Fast Traverse Block
 - 連続したバッファー・プール、PGESTEAL(NONE)
- zIIP処理の拡大
- スケーラビリティの向上
 - INSERTアルゴリズム2
 - ラッチ競合削減
 - EDMプール管理の向上
- お客様の課題への対応
 - バッファー・プールのシミュレーション
 - 実行時間の長いURについてのロック回避の改善

索引からデータへ： ランダム・キー・アクセス



- Fast Traverse Blockにノンリーフ・ページを収容
 - 制約：ユニーク索引で、サイズ64バイト以下
- デフォルトでオンになり、上限をユーザー指定
Index Memory Optimization Daemonプロセスが索引の使用状況をモニター
 - カタログ表からオブジェクト水準の指定も可能

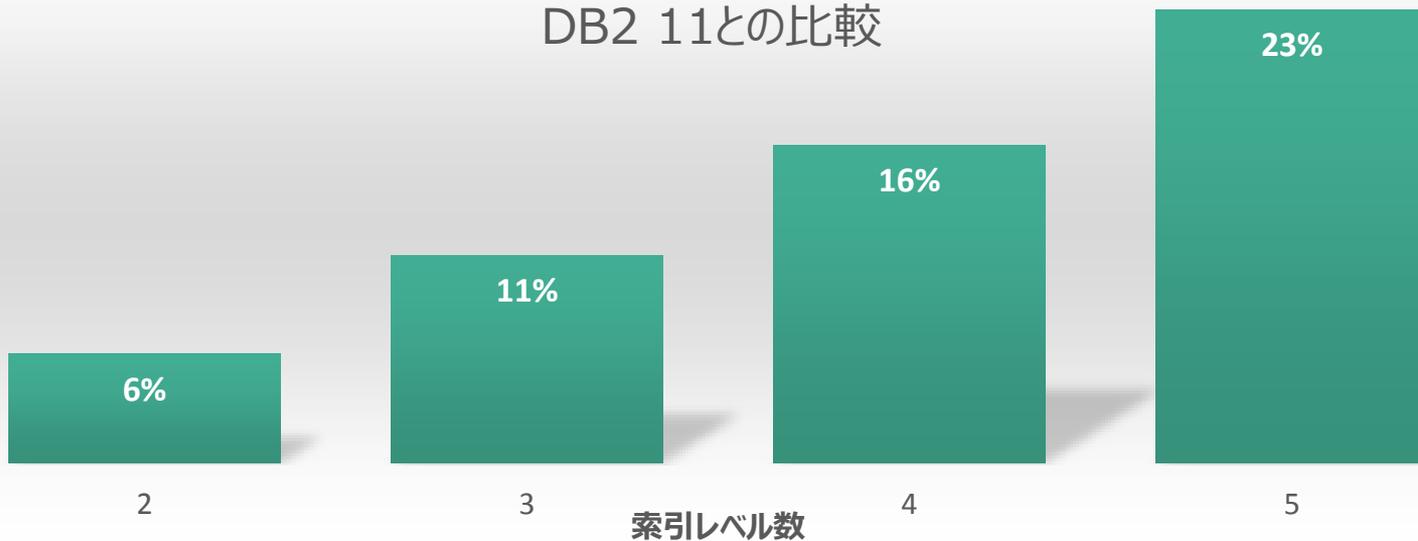
FTB(Fast Traverse Block)の挙動

- DB2システム・パラメーターINDEX_MEMORY_CONTROLによってオン/オフと上限を指定
 - AUTO、DISABLE、または500～200000 MB
 - デフォルトでは、バッファ・プール割り振りの合計量の20%までを使用
- インメモリ構造体を作る索引は、DB2が候補索引の使用状況をモニターして選択
 - 主にTraverseカウンターとその他で調整して制御
- ユーザーは、カタログ表(SYSIBM.SYSINDEXCONTROL)入力により、挙動の操作が可能（擬似削除エントリーのクリーンアップと同様）
- オブジェクトとストレージの使用状況をモニターする新しい表示コマンドが使用可能
- FTB使用量を記録するための新しいインフラストラクチャを追加

メモリー内索引の活用 – バッチSELECTの例

DB2 12単純ルックアップの向上幅(%)

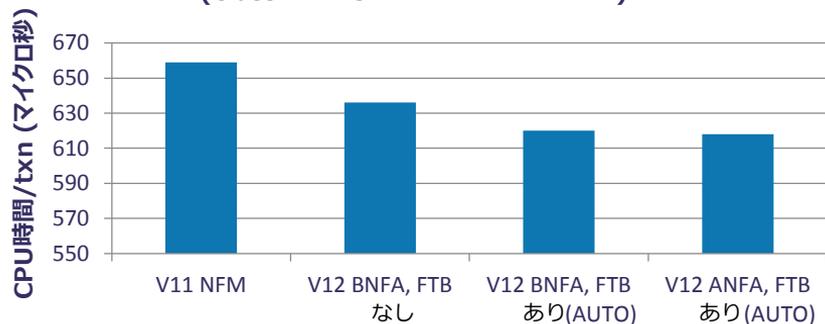
DB2 11との比較



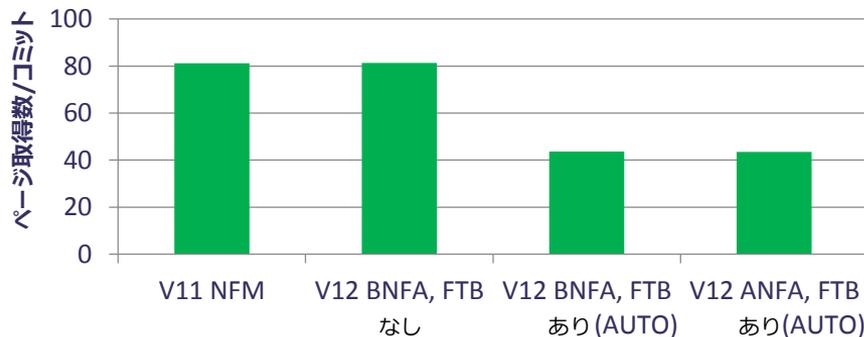
メモリー内索引の活用 - 2ウェイ・データ共有OLTP

DB2 V12 vs V11 - RELEASE(COMMIT)

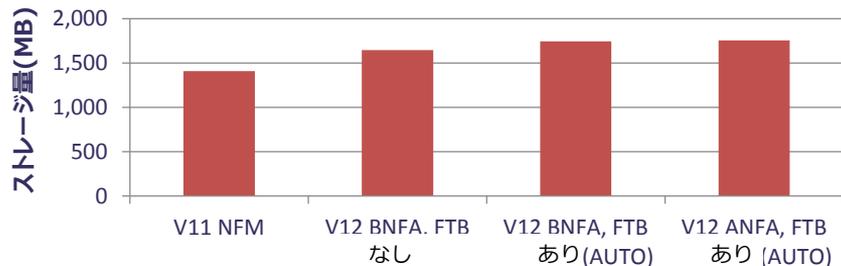
クラシックIRWW 2ウェイ: DB2 CPU時間/トランザクション
(Class2 + MSTR + DBM1 + IRLM)



クラシックIRWW 2ウェイ: ページ取得数/コミット



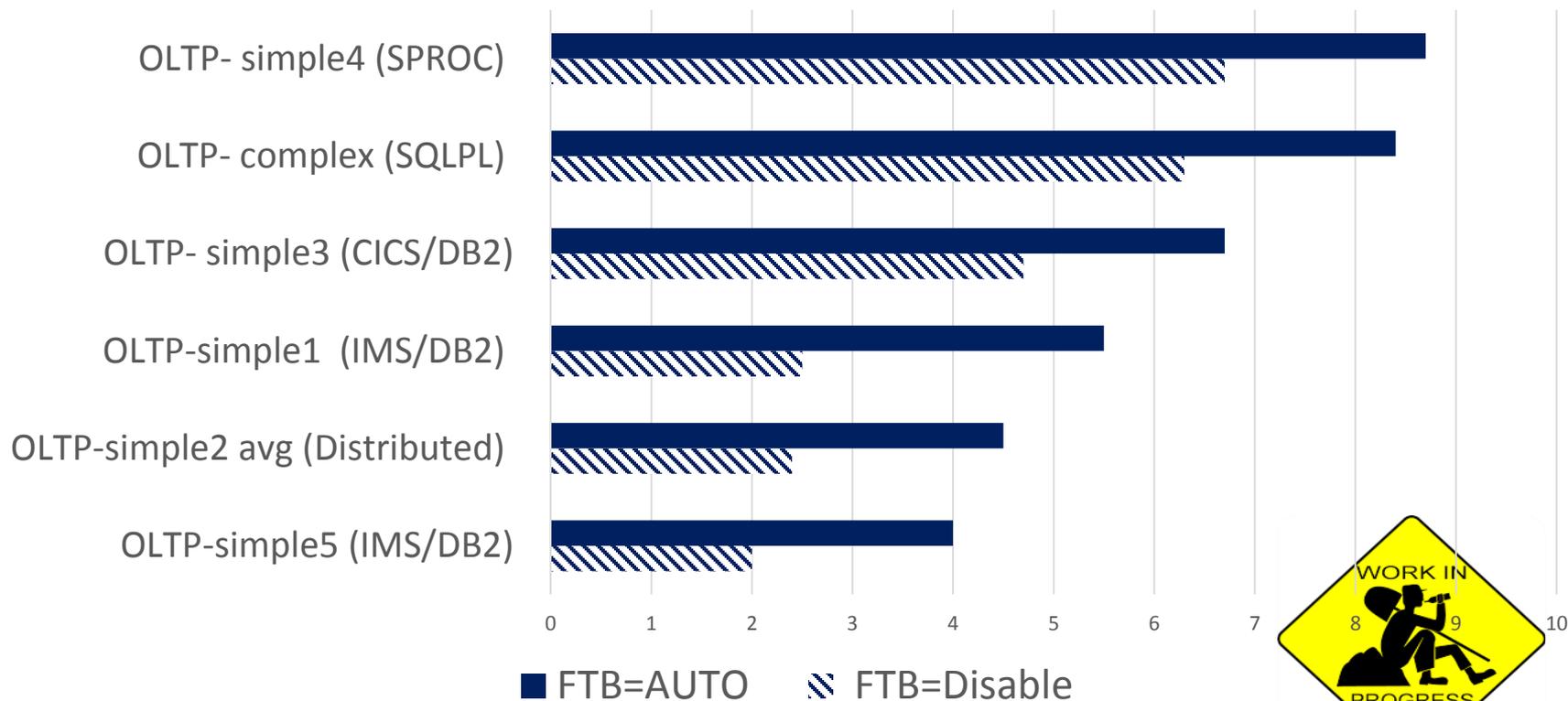
クラシック IRWWウェイ: DBM1実ストレージ量
ローカル・バッファ・プールのストレージを除く



所見

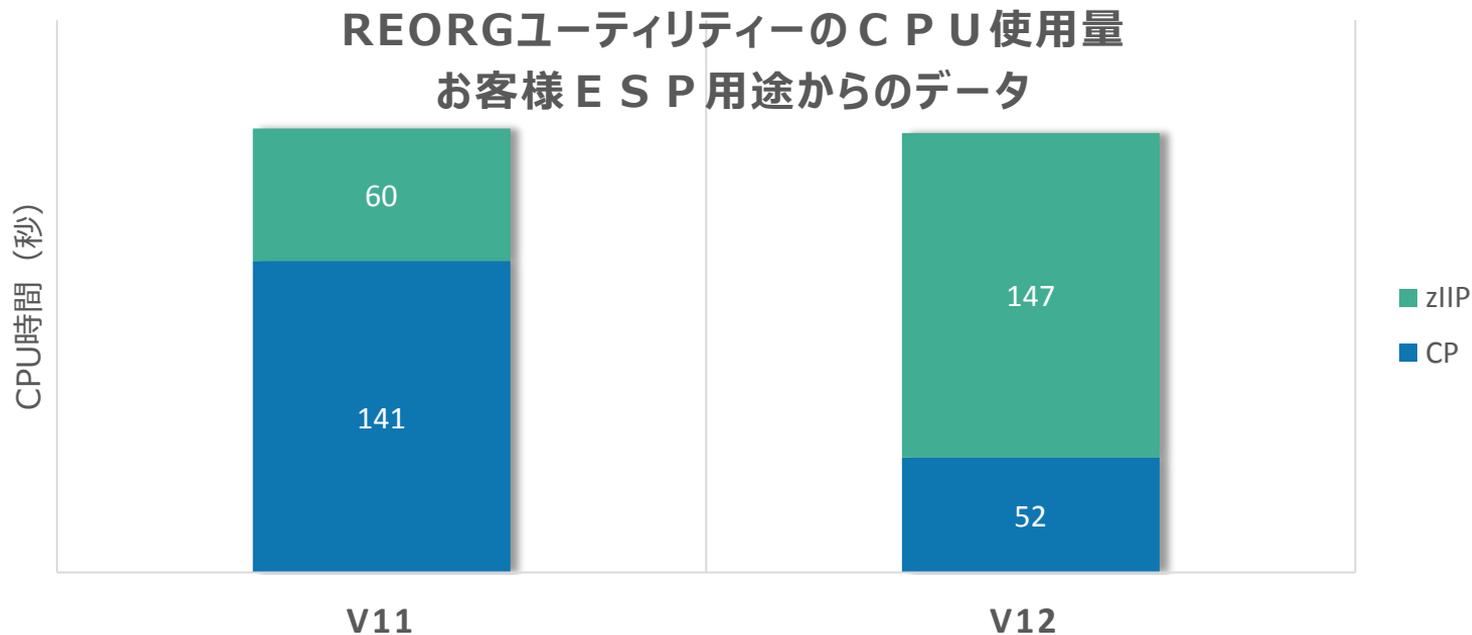
- FTB無しでも、平均約3.5%のDB2 CPU時間/トランザクション削減
- FTB使用により 3%の節約増
- ページ取得数45%削減
- 実ストレージ増加量は各メンバー300 MB

DB2 12 : OLTP ワークロードにおける DB2 11からの CPU 削減 (%)



zIIP関連の更新

- CPU並列照会 - 100% zIIP適格
- REORG、LOADユーティリティーのRELOADフェーズ



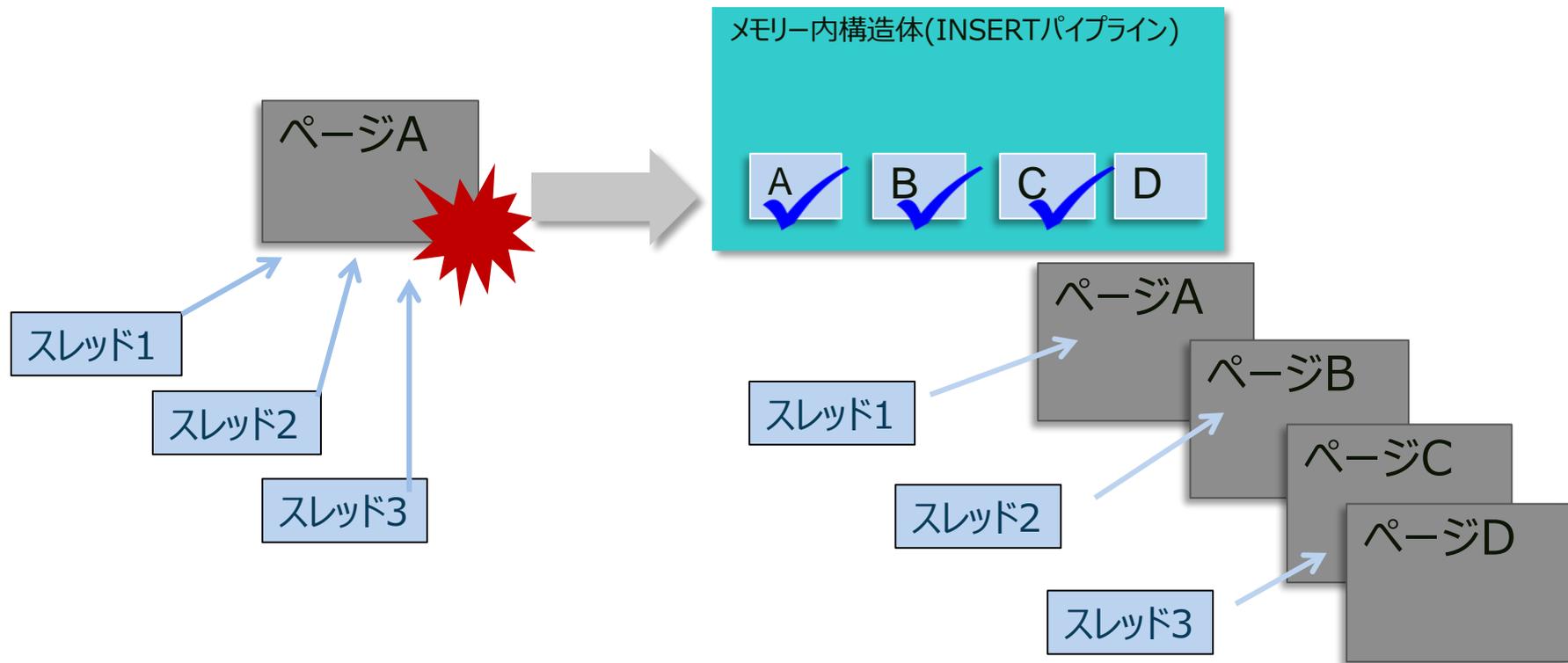
DB2 12の主要フィーチャー – システム・パフォーマンス



システム・パフォーマンス

- 大きな実メモリーの活用
 - メモリー内索引 = Fast Traverse Block
 - 連続したバッファー・プール、PGESTEAL(NONE)
- zIIP処理の拡大
- スケーラビリティの向上
 - INSERTアルゴリズム2
 - ラッチ競合削減
 - EDMプール管理の向上
- お客様の課題への対応
 - バッファー・プールのシミュレーション
 - 実行時間の長いURについてのロック回避の改善

新しいINSERTアルゴリズムの導入



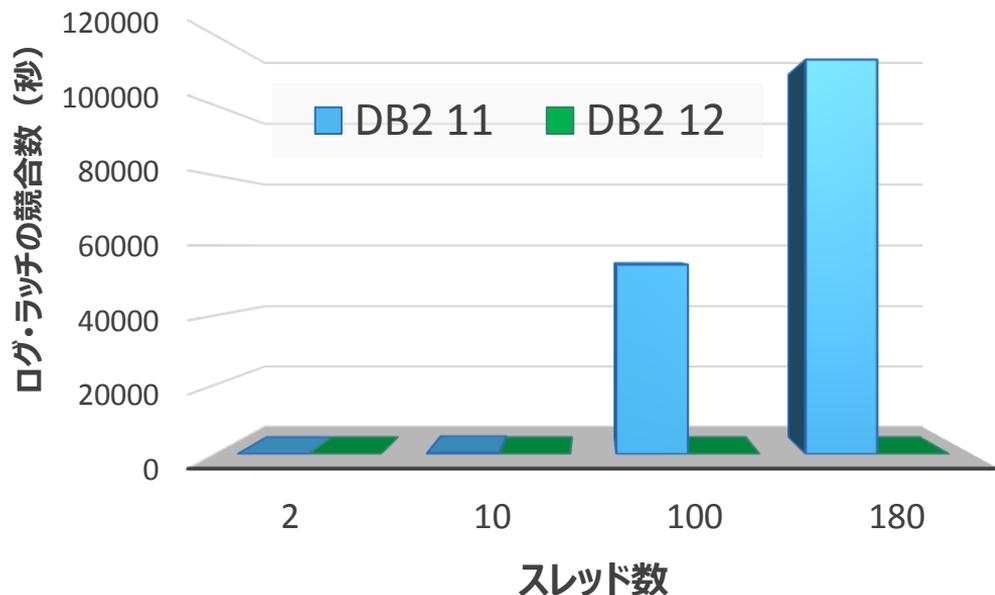
INSERTアルゴリズム2

- Member Clusterを指定したUTSが対象(APPEND有無とも)
- システム単位： DSNZPARAM: DEFAULT_INSERT_ALGORITHM
 - 1 - 基本INSERTアルゴリズムを使用(アルゴリズム1)
 - 2 - 高速INSERTアルゴリズムを使用(アルゴリズム2) - DB2 12 FL500ではデフォルト
- オブジェクト単位： CREATE TABLESPACEまたはALTER TABLESPACE
INSERT ALGORITHM節
 - 0 - サブシステム(メンバー)デフォルトを使用
 - 1 - 基本INSERTアルゴリズムを使用(DB2 11挙動)
 - 2 - 高度INSERTアルゴリズムを使用

DB2ラッチ削減

- DB2内部ラッチ競合の緩和
 - LC14 バッファーマネージャーラッチ
 - LC19 ログラッチ
 - LC23 ページラッチタイマー
 - LC23 ページアンラッチ
 - LC24 EDMラッチ

ログラッチの削減



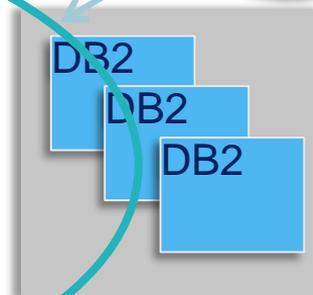
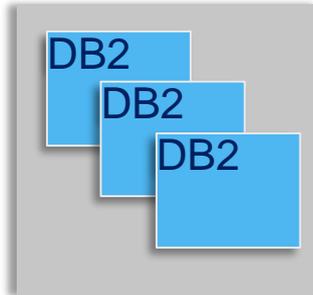
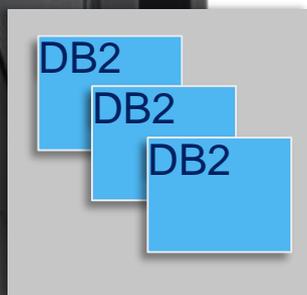
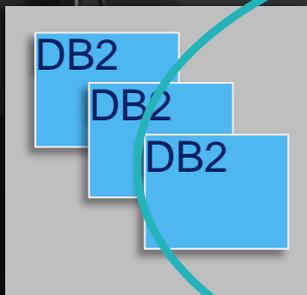
1000万INSERTを目標にしたベンチマーク環境

IBM

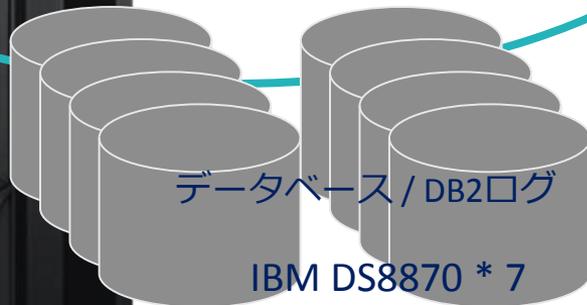
カップリング・ファシリティー

z13 2 ICF * 10 CPs, 200GB

zLinux上で
Javaクライアント
証券取引模擬アプリケーション
z13 2 zLinux * 18 CPs, 40GB



12 DB2 サブシステム
4 z/OS 占有 LPAR * 18 CPs
512 GB Memory z13



データベース / DB2ログ

IBM DS8870 * 7



1,170万 INSERT/秒

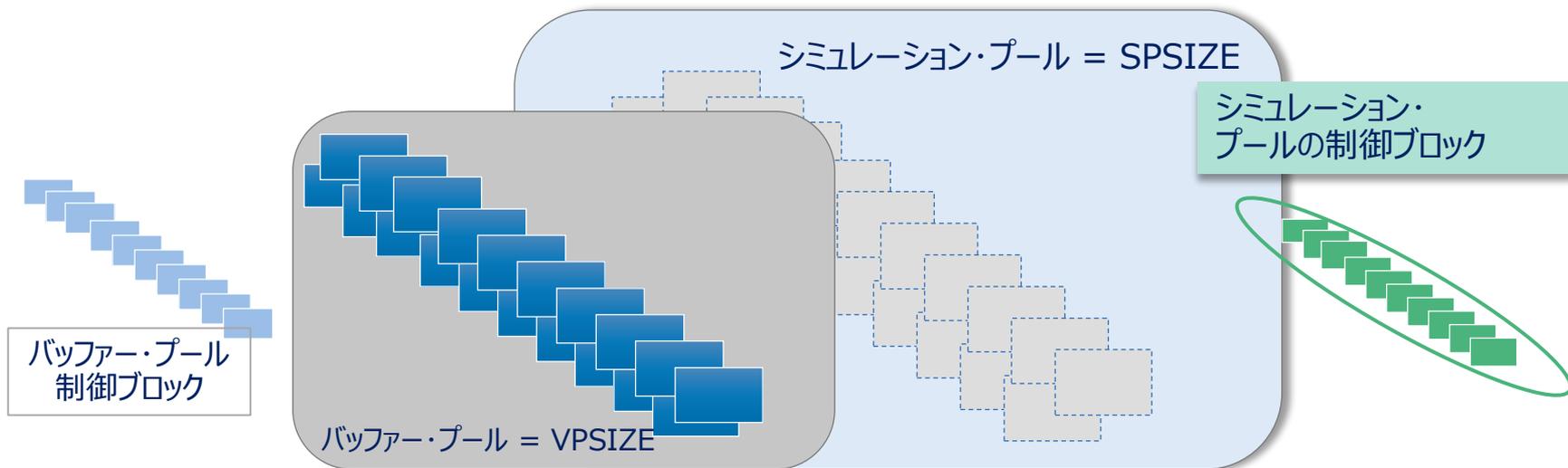
- 証券取引処理をシミュレーション
- DB2 12スケラビリティ・フィーチャーを活用
 - 新しいINSERTアルゴリズム、>4GBログ、ログ・ラッチ削減
- すべてをz13ボックス1筐体で処理
 - 12ウェイDB2データ共有、4ウェイSysplex
 - 2つのzLinuxで1200クライアントが同時稼動
 - 2000 Mバイト/秒のアクティブ・ログI/O
- 索引定義なしに12メンバーで1秒間あたり 1,170万回
- 区分索引を定義すると12メンバーで1秒間あたり 530万回

お客様から寄せられる課題への対応

- バッファ・プール・シミュレーション
- データ共有における実行時間の長い作業単位からの影響の低減
- データ共有におけるUNLOAD & RUNSTATSの影響の抑制
 - 新しいキーワードREGISTER NO/YESをUNLOAD / RUNSTATSに追加
- ピア・リカバリーのサポート
- 高速DRDAロード
- データ共有におけるDRDAグローバル・トランザクションのサポート
- IMS接続プーリング
- 静的SQLのためのRLF

バッファ・プール・シミュレーション(DB2 12、PI22091適用の11)

- バッファ・プールのサイズ増加の効果はワークロードに依存
 - データのアクセス・パターン、現状のバッファ・プール構成、その他
- バッファ・プール・サイズ増加の効果をも、ワークロードを実稼働させながら正確にシミュレーション



シミュレーションの使い方 - 例 :

- BP1のサイズを500,000増やし、現行設定VPSEQT = 60 を VPSEQT = 30 でBP1をシミュレーション
- 統計データを収集、またはDISPLAY BUFFER POOLコマンドで表示
 - SPSIZEを変えてシミュレーションを繰り返す場合は、いったんSPSIZE(0)でリセットが推奨
 - SPSIZEの値ごとに十分なサンプルを取得(推奨2~3時間)
 - 実行コスト・ストレージ : SPSIZE * 0.02、 CPU : シミュレーション・プールあたり約1%



シミュレーション出力

出力表示 - DIS BPOOL (BP1) DETAIL、または測定期間の
統計トレース・クラス1、2

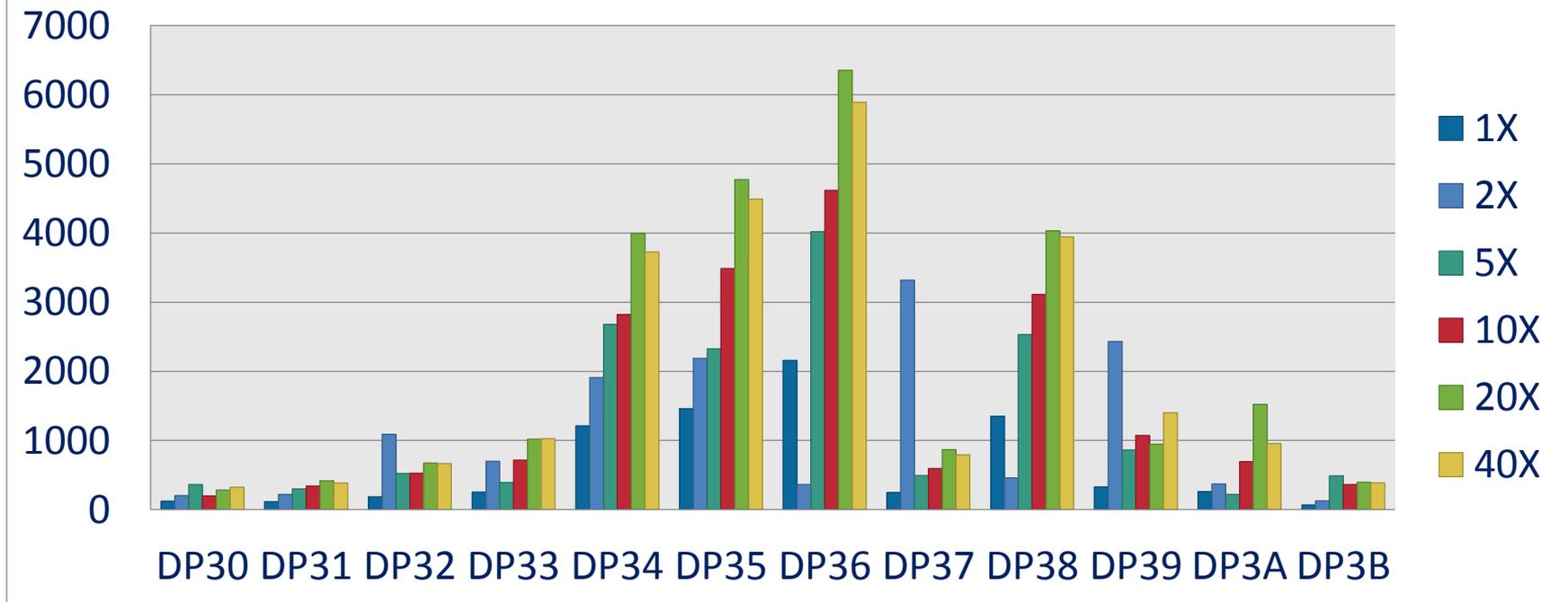
```
DSNB432I  -CEA1 SIMULATED BUFFER POOL ACTIVITY -  
          AVOIDABLE READ I/O -  
          SYNC  READ I/O (R)  =25463982  
          SYNC  READ I/O (S)  =81181  
          ASYNC READ I/O      =15470503  
          SYNC  GBP READS (R) =11172099  
          SYNC  GBP READS (S) =4601  
          ASYNC GBP READS     =1181076  
          PAGES MOVED INTO SIMULATED BUFFER POOL =53668641  
          TOTAL AVOIDABLE SYNC I/O DELAY =35321543 MILLISECONDS
```

同期I/O数 / 測定期間 = 回避可能な毎秒の同期I/O

- $(25463982 + 81181)/360 =$ 毎秒70958 I/O

お客様の結果例：メンバーあたりの回避可能I/O数

回避可能な毎秒の同期I/O



データ共用におけるロック・アボイダンス

現在の課題

- GBP DEPENDENT の 資源に対して実行時間の長大なURが存在するとデータ共用グループ全体にわたるロック・アボイダンスのチェックが損なわれる
- ロック・アボイダンスが効かずに不必要なロックやアンロック発生の可能性はある

DB2 12 での解決策

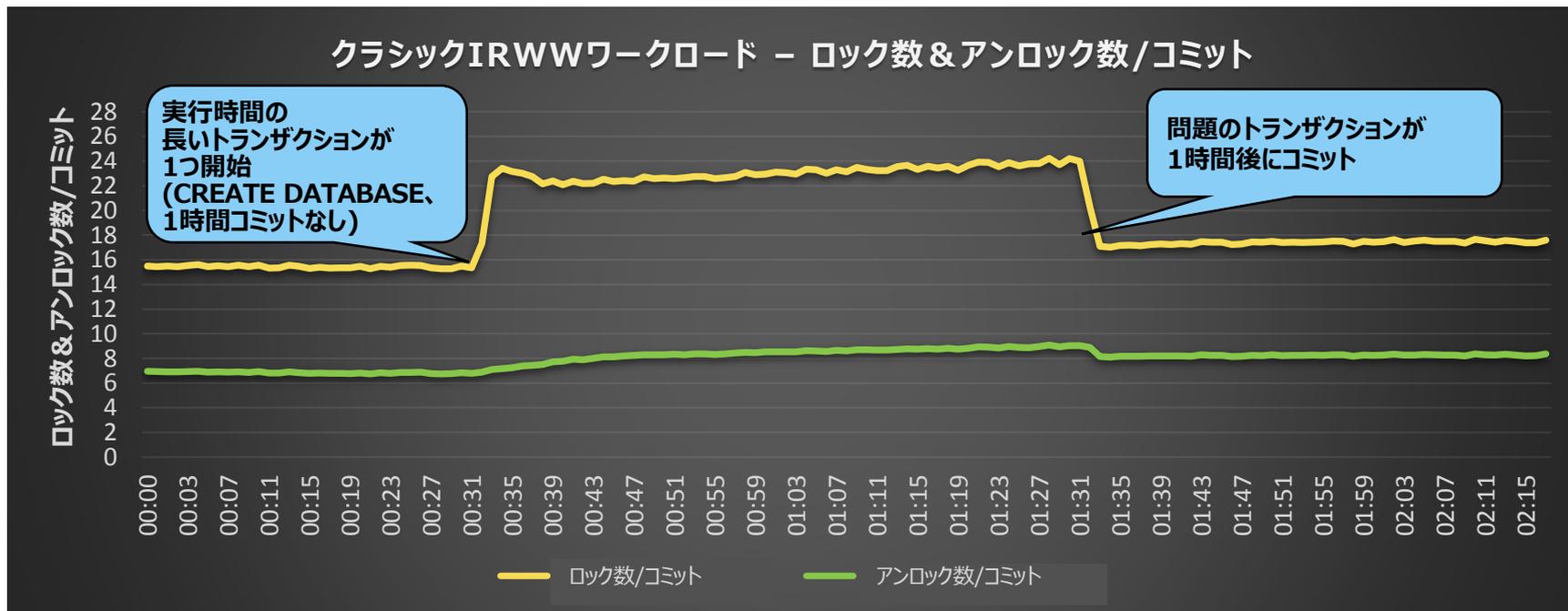
- これまでよりも粒度の小さなコミット&読み取りLRSN情報を維持して、問題を起こすURの影響に対処
- 実行時間の長大なURが存在してもデータ共用環境において、よりよいロック・アボイダンスでCPUを削減

コミットおよび読み取りのLRSNを調べる新しいコマンド

– DISPLAY DATABASE (dbname) SPACE (tsname) LRSN

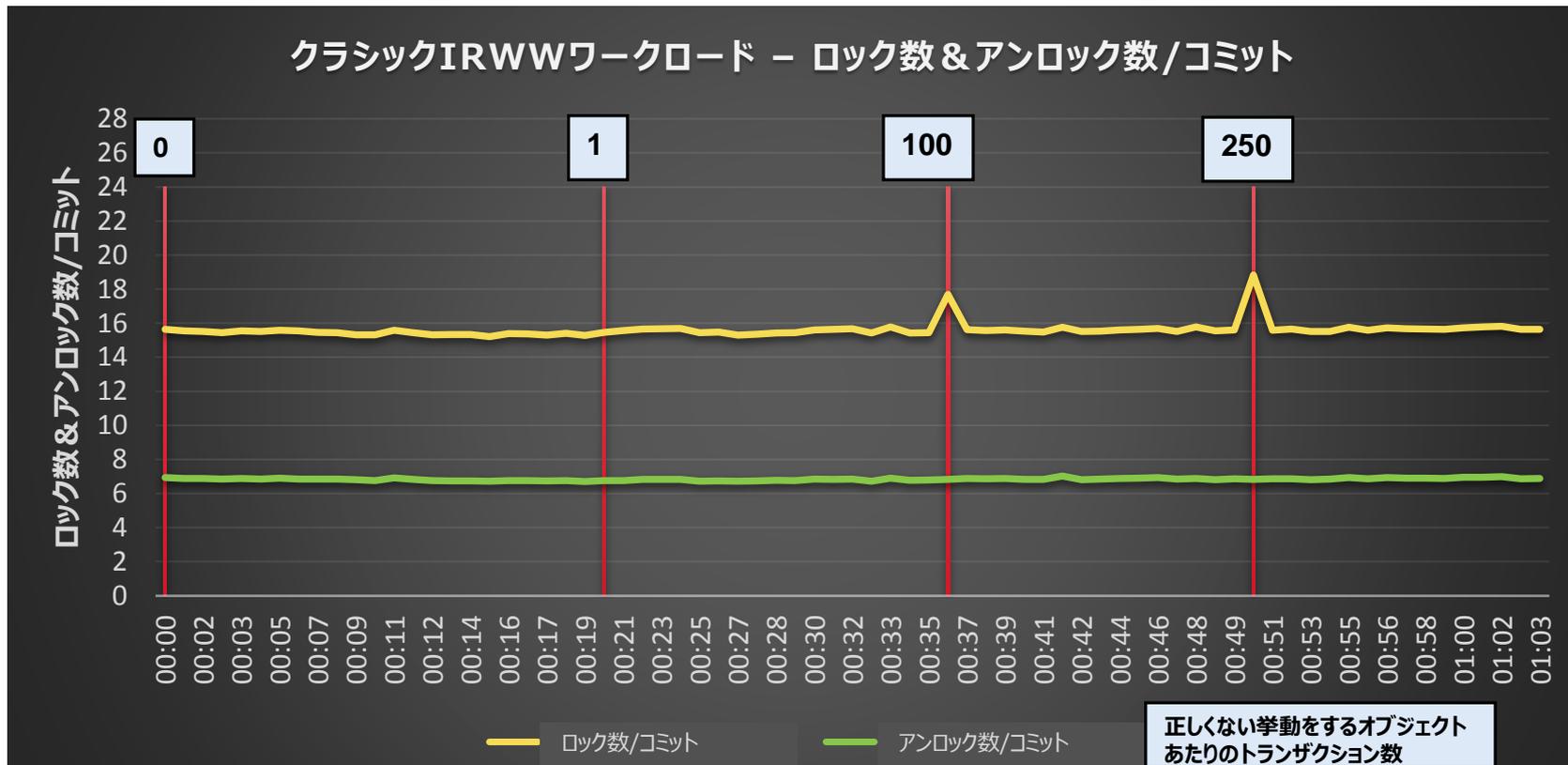
NAME	TYPE	PART	LRSN		TIMESTAMP
TSNAME1	TS	1	COMMIT	00CD534D63D8BA000000	6/18/2014 15:23:49
			READ	00CD535930B89F000000	6/18/2014 16:16:37

コミットLSN – 現在の使用状況(DB2 11)



- データ共有環境に挙動の悪いトランザクションが1つでもあると、ほかのトランザクションのロックが増加
- この2ウェイ・クラシックIRWWワークロードの例では、そのようなトランザクション1個によってロック回避の働きが損なわれ、DB2クラス2 CPU使用量を7.6%増加

コミットLSN – DB2 12



DB2 12の主要フィーチャー – SQL照会パフォーマンス



SQL パフォーマンス

- アクセス・パス安定化
 - 動的SQLプラン安定化の新サポート
 - 静的プラン安定化の使いやすさの向上
 - オプティマイザーによる統計プロファイルの自動更新
- 照会パフォーマンスの向上
 - UNION ALL、OUTER JOINの強化
 - 広範なアクセス・パスの向上
 - ランタイム処理の向上
 - UDFキャッシュ
 - ソートの向上
 - 適応索引選択

動的SQLステートメントの安定化

動的プランの問題点

- 繰り返し使う動的SQLステートメントのパフォーマンスが変動する
- 環境の変化によってアクセス・パスが変化し、パフォーマンスが退行し、その管理が簡単にできない
 - RUNSTATS
 - SWメンテナンスの適用
 - DB2のリリース移行
 - zparmの変更
 - スキーマの変更

DB2 12の解決策

- 動的SQLの安定化 (Stabilize Dynamic SQL)
 - システム・パラメーター
CACHEDYN_STABILIZATION
- ステートメントをカタログにキャプチャー
 - SYSIBM.SYSDYNQRY
 - モニター有無指定コマンド
 - ステートメントをフィルターする閾値
 - ステートメントのグループ化
- SDQ (stabilized dynamic queries) のEXPLAIN
- FREEによるSDQの削除
- SDQの無効化、再有効化

動的SQLステートメントの安定化

動的SQLステートメント

SQL 1
SELECT cols FROM table ...

PREPARE

EXECUTE

動的ステートメント・キャッシュ

```
SQL 1 : SELECT .. EXEC 1000  
SQL 2 : UPDATE .. EXEC 50  
SQL 3 : SELECT .. EXEC 2000  
SQL 4 : SELECT .. EXEC 100
```

STABILIZE

```
-START  
DYNQUERYCAPTURESTBLGRP (APP1)  
THRESHOLD (200) CURSQLID (APP1ID)  
MONITOR (NO/YES)
```

SYSIBM.SYSDYNQRY
SDQ APP1 SQL 3
SDQ APP1 SQL 4

動的SQLステートメントの安定化

動的SQLステートメント

SQL 1
SELECT cols FROM table ...

EXECUTE

動的ステートメント・キャッシュ

SQL 5 : SELECT .. EXEC 100
SQL 7 : UPDATE .. EXEC 50

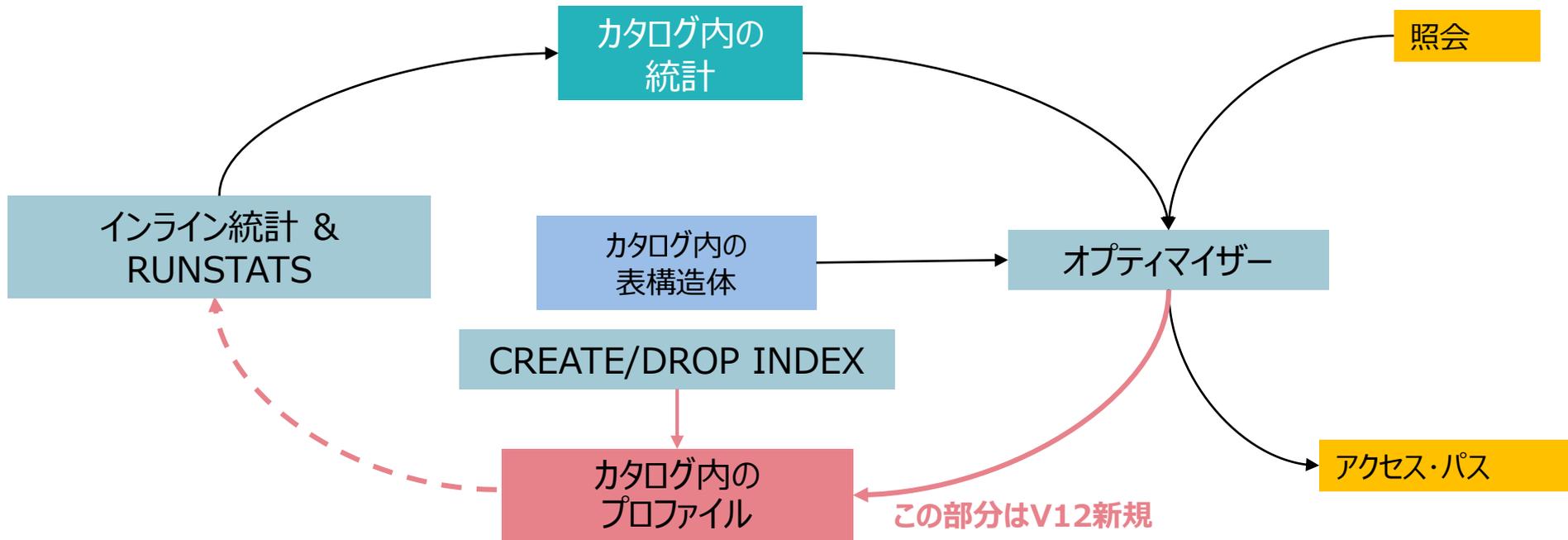
SYSIBM.SYSDYNQRY
SDQ APP1 SQL 3
SDQ APP1 SQL 4

EXPLAIN STABILIZED
DYNAMIC QUERY STMTID 3
FREE STABILIZED DYNAMIC
QUERY SDQSTMTID(1)

RUNSTATSの機能強化 – SQLパフォーマンス

- アクセス・パス選択向上のための RUNSTATSの拡張
 - クラスタ率の式を改良
 - ステートメント・キャッシュの無効化
 - RUNSTATSではオプション化
 - ほかのユーティリティでは、実行前にオブジェクトが保留状態のときのみ
 - インライン統計のプロファイル・サポート
 - FREQVALのCOUNT自動設定
 - 偏りのある値での収集をDB2に任せる
 - 偏りのない値になるまで収集、最大で上位100まで
- オプティマイザーがRUNSTATS推奨で統計プロファイルを自動更新
 - DSNACCOXは、プロファイルに更新があるとRUNSTATS実行を推奨
 - RUNSTATSでUSE PROFILEを指定すると、更新された推奨に従って統計を収集

RUNSTATSプロファイル管理の機能強化



照会パフォーマンスの向上

- UNION ALL、OUTER JOINの強化
 - ソート、マテリアライズの回避
 - 外部結合表の再順位付けによるマテリアライズの回避
 - UALレグまたはOUTER JOIN照会ブロック内の述語の取り込み
 - 外部部分のソートによる内部部分の順次アクセスの確保
 - マテリアライズが必要になった場合の、作業ファイル使用のバイパス
 - ORDER BY、FETCH FIRSTのUALレグへの取り込み
 - 必要のない列、表の切り落とし
- ランタイム最適化
 - 適応索引による実行中のパス調整
 - UDFに同じ入力に関わる場合のUDFキャッシュ
 - SORT処理向上
- 述部とオプティマイザーの機能強化

例：表の結合 - アーカイブ・トランスペアレンシー

元の照会

```
SELECT T1.C1, T2.C2 FROM T1 LEFT JOIN T2 ON T1.C1=T2.C1;
```

DB2 V11による書き換え

```
SELECT TX1.C1, TX2.C2 FROM (  
SELECT * FROM T1  
UNION ALL  
SELECT * FROM H1) AS TX1  
LEFT JOIN (  
SELECT * FROM T2  
UNION ALL  
SELECT * FROM H2) AS TX2  
ON TX1.C1 = TX2.C1;
```

マテリアライズ(全列、全行)

マテリアライズ(全列、全行)

作業ファイルのJOIN

例：表の結合 - アーカイブ・トランスペアレンシー

元の照会

```
SELECT T1.C1, T2.C2 FROM T1 LEFT JOIN T2 ON T1.C1=T2.C1;
```

DB2 V12による書き換え

```
SELECT TX1.C1, TX2.C2 FROM (  
  SELECT T1.C1 FROM T1 UNION ALL  
  SELECT H1.C1 FROM H1) AS TX1  
LEFT JOIN TABLE(  
  SELECT T2.C1 FROM T2  
  WHERE T2.C1 = TX1.C1  
  UNION ALL  
  SELECT H2.C1 FROM  
  WHERE H2.C1 = TX1.C1) AS TX2  
ON 1=1;
```

パイプライン化

JOINを基本表に下げて、
内部部分をパイプライン化

作業ファイルなし

オプティマイザーは以下を
実施する可能性もある

- 内部部分の順次アクセスのため外部のソートを選択
- あるいは、各インナー・レグに対して疎索引(sparse index)を作成

ランタイム適応索引

- 汎用の検索照会は、フィルタリングが変わることが課題

```
SELECT * FROM CUSTOMER
```

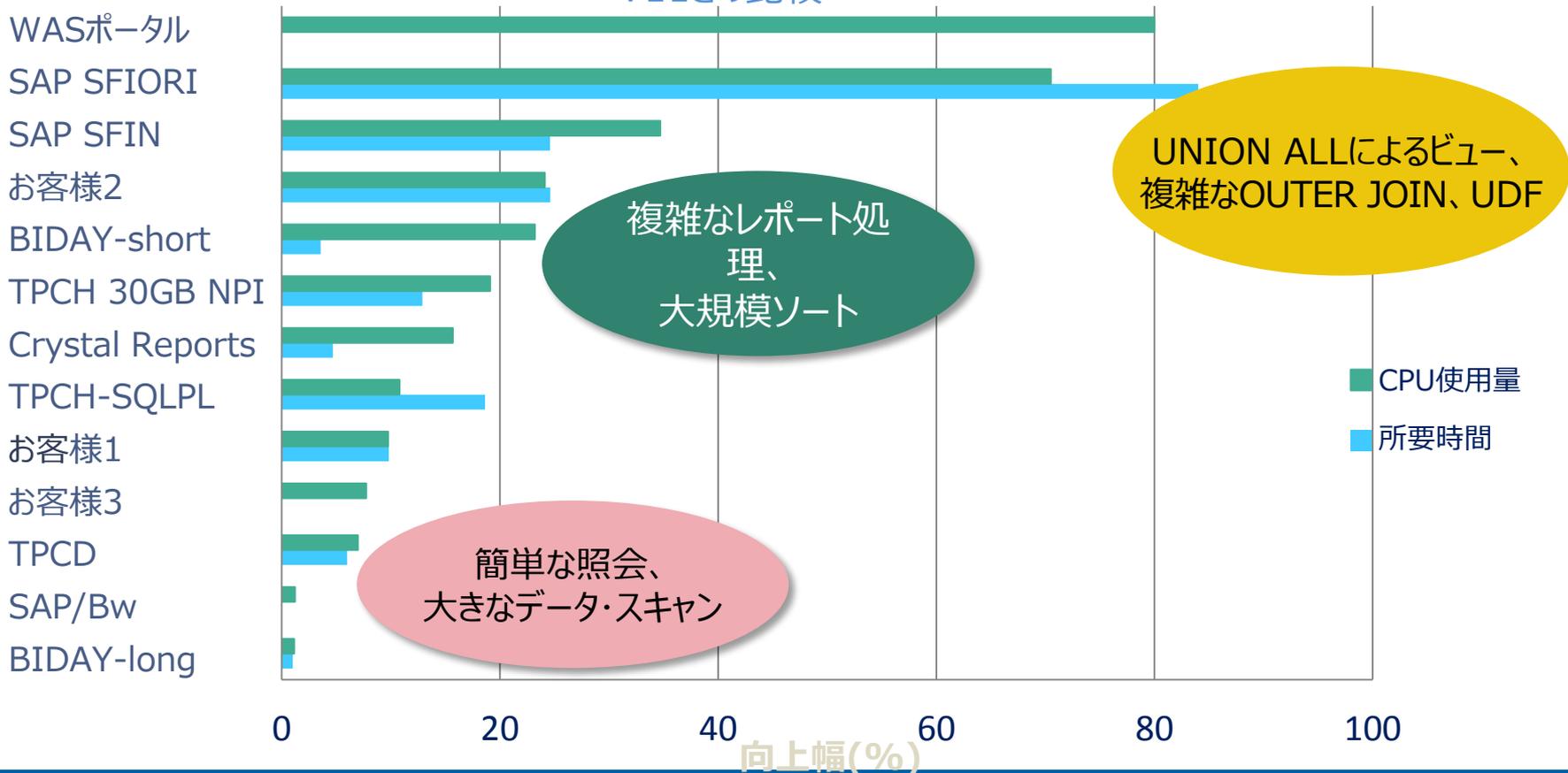
```
WHERE LASTNAME LIKE ? AND FIRSTNAME LIKE ? AND ADDRESS LIKE ?
```

```
AND CITY LIKE ? AND ZIPCODE BETWEEN ? AND ?
```

- リスト・プリフェッチ・ベースのプランで、索引フィルタリングを素早く決定(単一索引、複数索引)
- 求めたフィルタリングに基づいて、実行時に調整(適応)

V12照会ワークロードでのCPU使用量、所要時間の向上幅 (%)

V11との比較



DB2 12 ESP (早期サポート・プログラム) パフォーマンス情報

お客様	結果
お客様-1	Mixed : バッチで最大9% クラス2 CPU削減. CICSワークロードで zIIP時間が増加 (PMR対応). FTB enabled は disabled設定より大きく向上. UNION ALL は大きな向上を示した.
お客様-2	Positive : SAPワークロードをテスト. 経過時間13%向上. CPU時間16%削減. zIIP時間17%削減.
お客様-3	Mixed : リグレッションにおいてはCPU削減はみられなかった. REORGはGCP使用削減(zIIPオフロード)がみられた.
お客様-4	Positive : V12 において最大10%クラス2 CPU削減. FTB enabledにより、CPUの最大消費業務が大きな向上を見せた. REORG と LOAD は極めて大きなCPU使用削減がみられた.
お客様-5	Positive : 本番模擬環境において最大10%のCPU削減.
お客様-6	Mixed : ほとんど同等. アクセス・パスの変更に伴う悪化もいくつかみられた. DRDA Fast load により従来30分の処理が30秒に.
お客様-7	Positive : 初期導入で平均15%パフォーマンス向上. ANFAに移行後はさらによい結果がえら得た - バッチパフォーマンスは25-30% 向上した.

DB2 12の主要フィーチャー – アナリティクス、クラウド、モバイル



クラウド、
モバイル・アプリケーション

- IDAA機能の向上
 - アクセラレーター専用表
 - テンポラル表のサポート
 - インクリメンタル更新の向上
 - 移動中データの暗号化
 - SQLサポートの拡大
- 新しいアプリケーションのSQLサポート
 - z/OSMFのサポート
 - RestAPIのサポート
 - 配列の機能強化
 - 断片的変更
 - SQLページネーション
 - JSONサポートの機能強化
 - XML機能の向上
 - 複雑なトリガー
 - テンポラルの向上
 - マージの拡張

新しいアプリケーションの開発

- 近年のアプリケーション開発者の多くはRESTサービスやJSONフォーマットを使用
- DB2 Adaptor for z/OS Connectが、その手段を用意
- DB2 12とともにDB2 10、11でも提供

`http://10.25.14.17:259/service/db1/getcust?32537`

RESTful

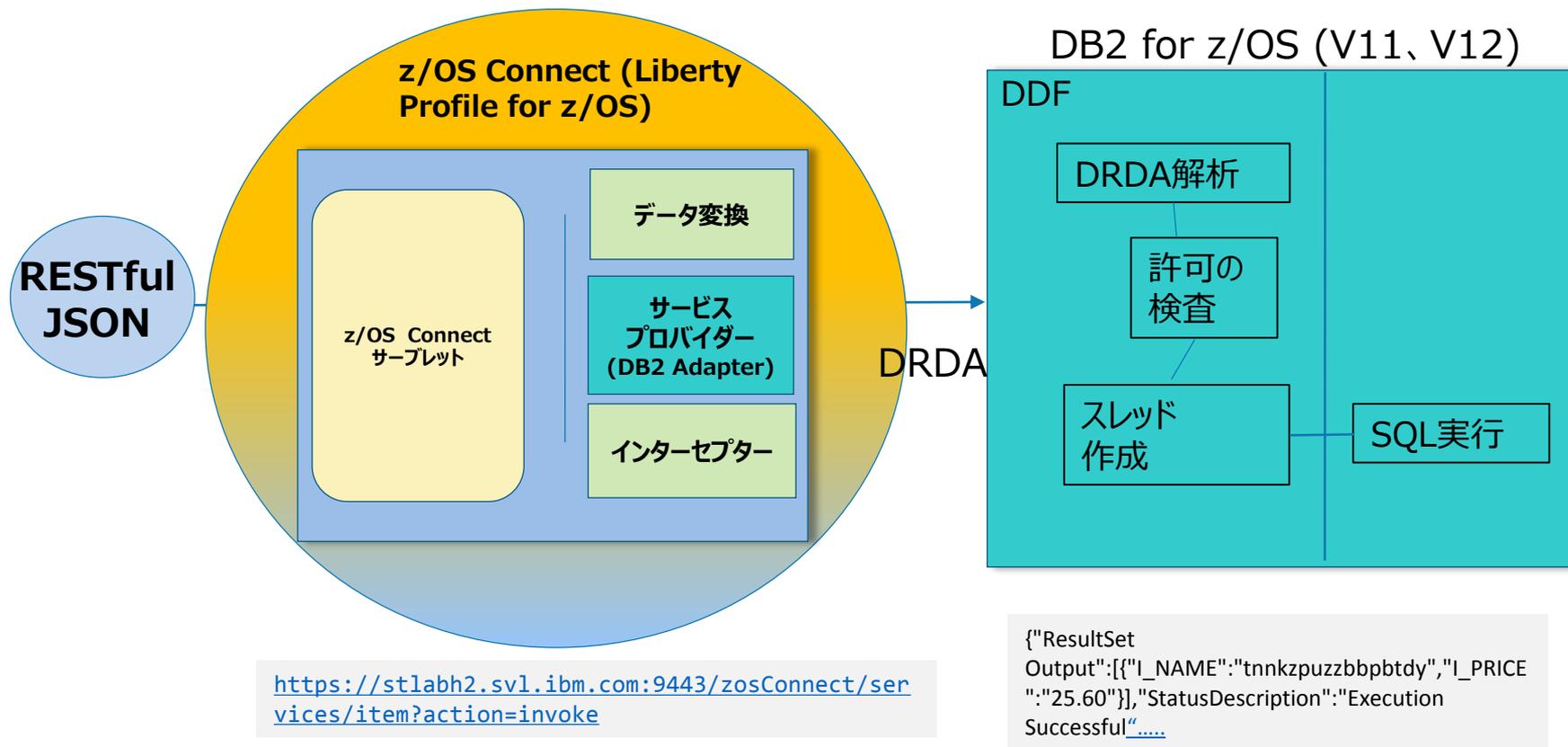


DB2

```
{  
  "customer":  
    {"ID": "32537",  
     "Name": "Bob Rady Enterprises"}  
}
```

JSON

DB2でのz/OS Connectを通じたRESTfulサービスの展開

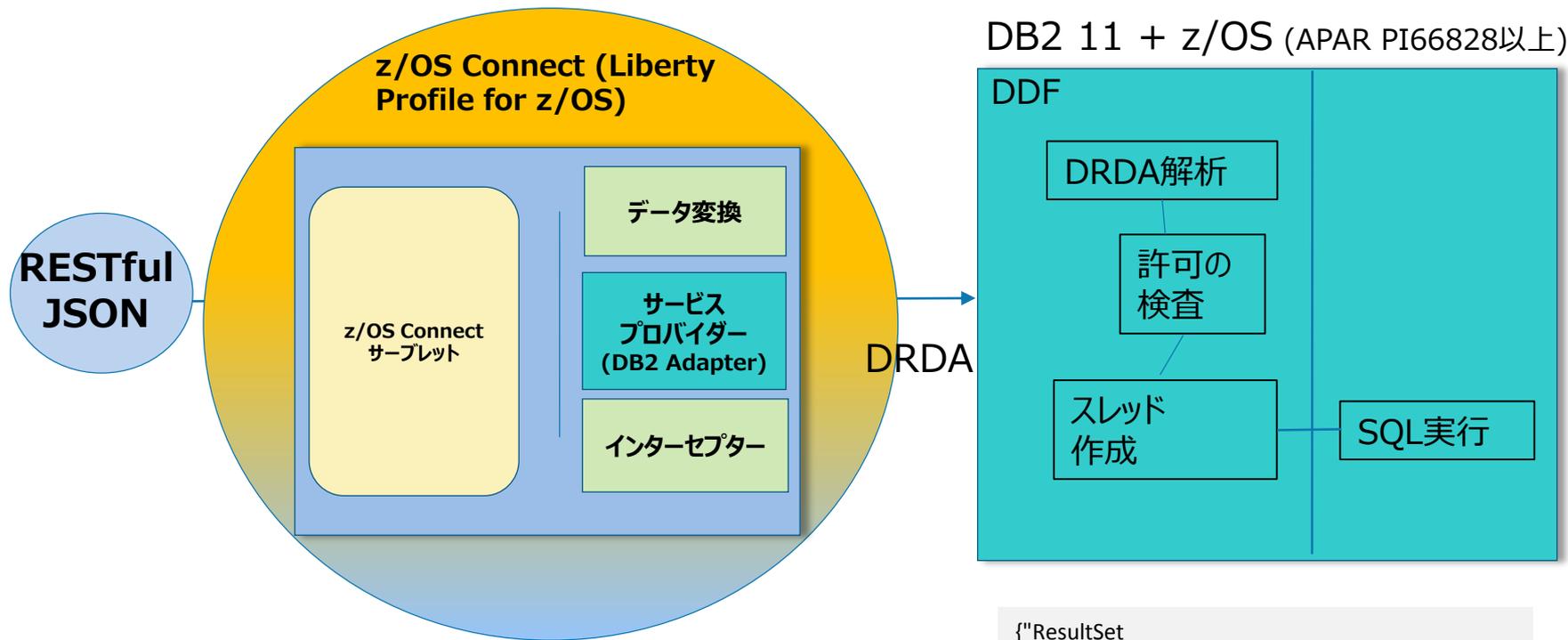


db2zアダプターのスケーリング – 最大106,000トランザクション/秒



- Db2zアダプターは、一定した直線状のスケーリングで106.8K tx/sに到達

ネイティブのRESTful API



<https://stlabh2.svl.ibm.com:9443/zosConnect/services/item?action=invoke>

```
{"ResultSet  
Output": [{"I_NAME": "tnnkzpuzzbbpbdy", "I_PRICE": "25.60"}], "StatusDescription": "Execution Successful".....
```

SQLページネーションの機能強化

- Webやモバイル分野の成長とともに、アプリケーション開発者には、良質のアプリケーションを開発するための、より効率的な方法が必要

- 数値によるページネーション

```
SELECT * FROM tab OFFSET 10 ROWS FETCH FIRST 10 ROWS ONLY
```

- データ依存のページネーション

従来の構文

```
WHERE (LASTNAME = 'SMITH' AND FIRSTNAME >= 'JOHN') OR (LASTNAME > 'SMITH')
```

新しい等価の構文

```
WHERE (LASTNAME, FIRSTNAME) > (SMITH, JOHN)
```

データの断片的変更

- 数百万行に影響する可能性のある単純なステートメントが及ぼすロックやログの影響を解消。
 - 例： “DELETE FROM T1 WHERE C1 > 7”
- 解決策
 - DELETEステートメントにフェッチ節を追加を許可。



```
DELETE FROM T1 WHERE C1 > 7 FETCH FIRST 5000 ROWS ONLY;  
COMMIT;
```

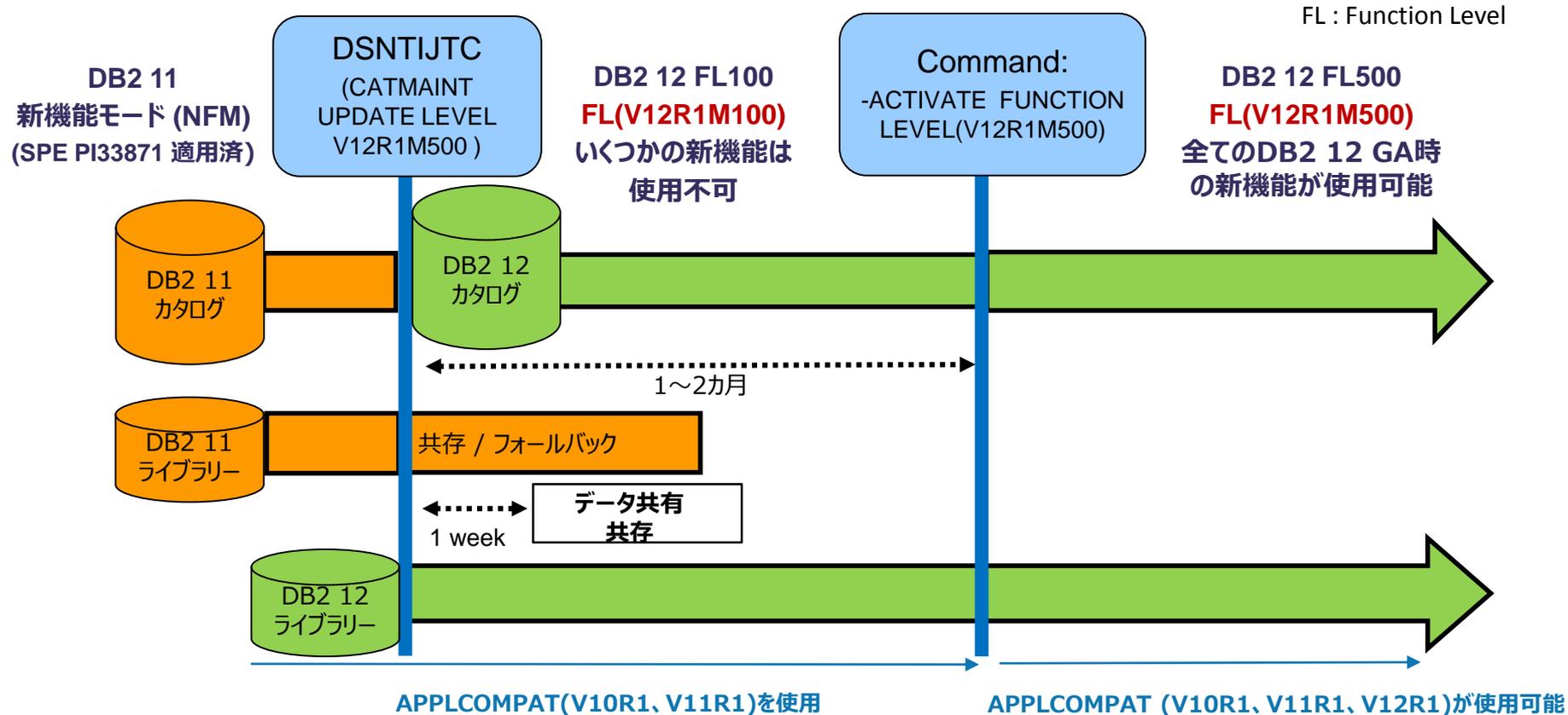
DB2 12の主要フィーチャー – DBA、連続可用性



DBA、365日24時間の
可用性

- 移行プロセスの簡易化
- オンライン・スキーマの拡張
- PBR相対ページ番号
- zEDCによるLOB圧縮
- 非同期LOCKストラクチャー二重化
- プロファイルの拡張
- セキュリティーの向上
- ユーティリティーの向上

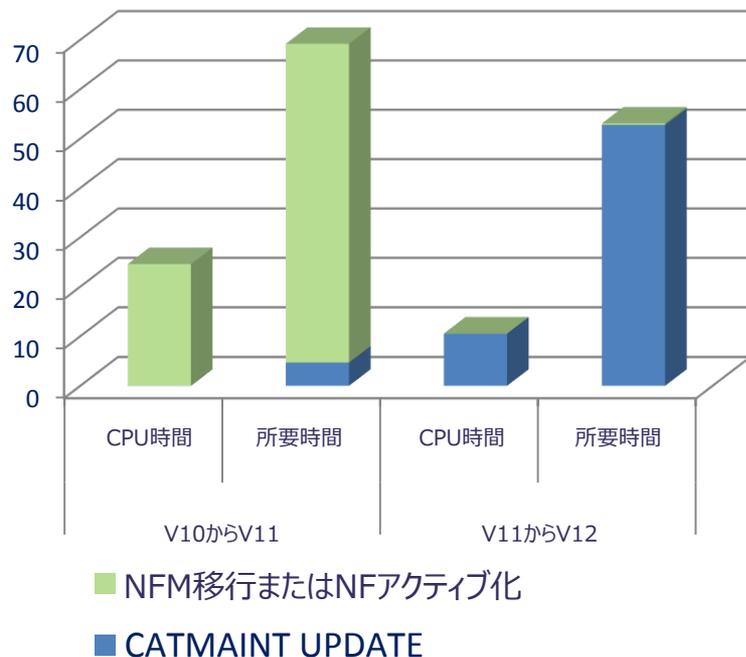
移行の簡易化



DB2 11 -> 12移行

- 1ステップのカタログ変更とオンライン・コマンドACTIVATE FUNCTION LEVELによる、簡易化された移行プロセス
 - APPLCOMPAT、フォールバックについては従来どおり
- BSDS変換による10バイト・ログRBAサポートが前提
- テンポラルRTS表
 - カatalogで定義し、使用可能化はオプション
- 移行パフォーマンスの比較
 - DB2 11 CATMAINT UPDATE + NFM移行
 - DB2 12 CATMAINT + 新機能アクティブ化(FL500)
 - 複数のお客様のカタログを使用(6 GB~28 GB)
 - 20~60%所要時間削減
 - 50% CPU時間削減

お客様のカタログを使った測定例 (サイズ23.8 GB)



オンライン・スキーマの機能強化

- パーティションの挿入
- オンライン据え置きによるALTER INDEX COMPRESS YES
 - 従来は索引がRBDP状態になった
- 列レベルのALTERの据え置きオプション
 - オンラインREORGによって実行
 - 可用性の制約 & ほかの据え置き変更との競合を回避

パーティションの挿入

従来の問題

- 大規模な範囲パーティション表 (Partition By Range) は、しばしばホット・スポットを生じ、ほかのパーティション全体に再均衡させるのは大きな労力を伴う
- 過去に選んだパーティション方式が、望ましくなくなっていることがある

DB2 12の解決策

- 限界キー値を指定して新しいパーティションを挿入する機能
 - ALTER TABLE ADD PARTITION ENDING AT xxx
- 既存のパーティションを分割して、新旧パーティションにデータ振り分け
- 変更保留によるオンライン変更
- REORGは、影響を受けたパーティションにのみ実施
- REORGの前に、PITリカバリーが不要

大規模パーティション表スペース: 相対ページ番号

従来の問題

- 表サイズの制約
- パーティションの最大数が、DSSIZEとページ・サイズにも依存
 - DSSIZE = 256 GBで、ページ・サイズ = 4Kの場合、パーティション最大数64
- DSSIZEを変更すると、表スペース全体にREORGが必要
- DSSIZEは表スペース・レベル

DB2 12の解決策

- 絶対ページ番号に換わる相対ページ番号 (RPN)
- パーティション数とパーティション・サイズ間の依存関係を解消
- 7バイトのRID (2バイトのパーティション番号、5バイトのページ番号)
 - 1つの表で最大280兆行
 - 最大1 Tbのパーティション・サイズをサポート
- DSSIZEは1~1024 GBの範囲で増加
- DSSIZEはパーティション・レベル

範囲パーティション表(Partition By Range) – 絶対 vs 相対

4Kで最大16 TB、
1.1兆行

最大12ビットの パーティション番号	20ビット、またはそれ以上のページ番号	8ビットのマップ・ エントリーID
000000000011	000000000000000000111	00000101

パーティション番号に使うビット数に応じて境
界が移動

境界固定

4Kで最大4 PB、
最大280兆行

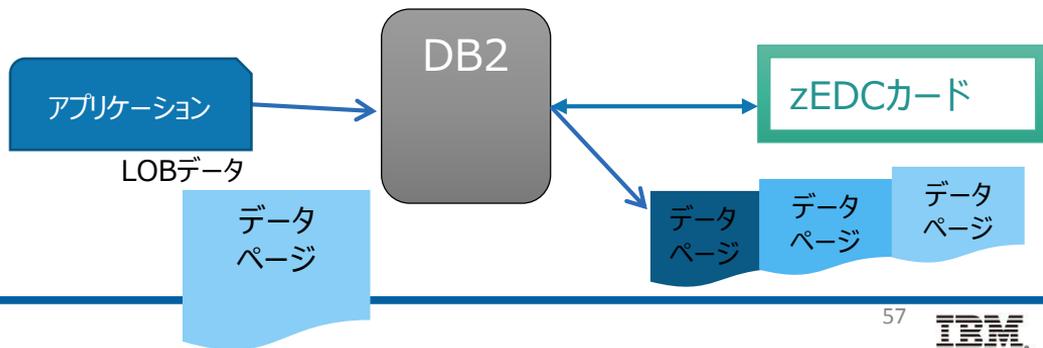
16ビットのパーティション番号 (ヘッダー・ページに格納)	32ビットのページ番号(データ・ページに格納)	8ビットの マップ・エントリー
0000000000000011	00000000000000000000000000000000111	00000101

境界固定

境界固定

zEDC ExpressによるLOB圧縮

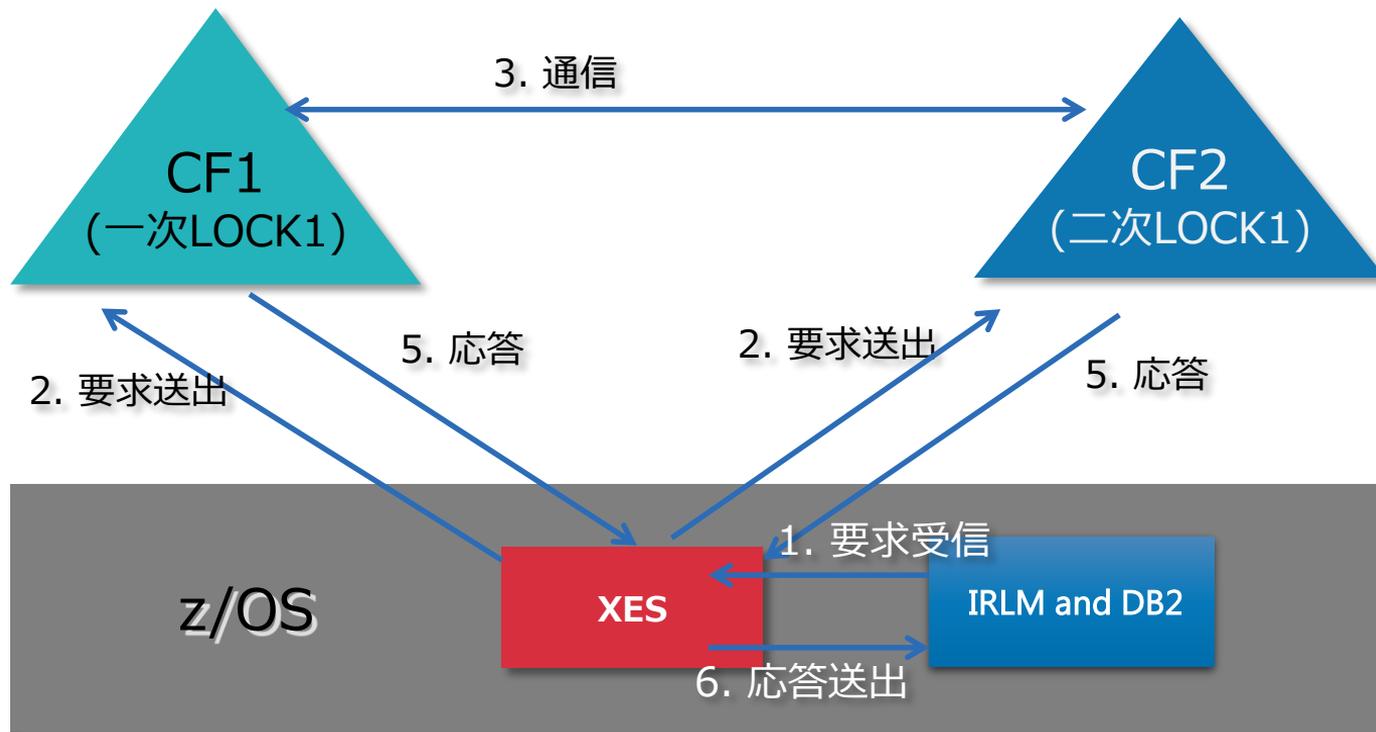
- LOB圧縮をzEnterprise Data Compressionカードによって断片ごとに処理
 - 断片はLOBの全体または1Mまでの一部分
- DB2 12 FL500 (新機能アクティブ化後)が必要
 - UTSの基本TSでLOB TS COMPRESS YES
 - SYSSPUXA、SYSSPUXB、SYSDBDXAのDB2ディレクトリーLOBに適用可
 - 新しいシステム・パラメーターCOMPRESS_DIRLOB
- 圧縮の対象
 - 非圧縮のテキスト文書、XML、HTML、非圧縮のPDF、Wordファイル、その他
 - JPGは元々圧縮済み



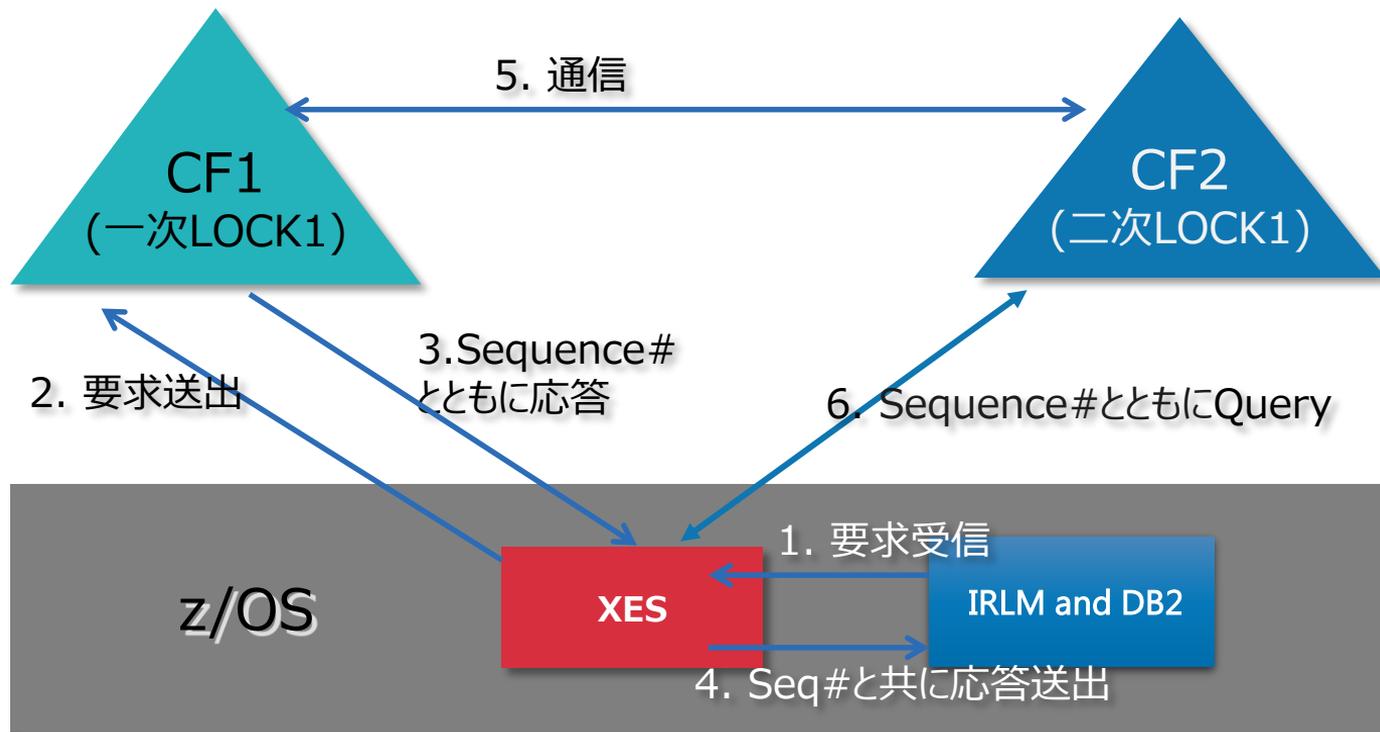
非同期CFロック二重化

- CF上のLOCKストラクチャーのシステム管理二重化の負荷を削減
 - 二次ストラクチャーの更新を一次側の更新と非同期に実施。
 - DB2はz/OSと連携しデータ整合性を確保。すべての更新ロックは、対応するundo/redoレコードがDASD上のDB2アクティブ・ログに書かれる前に、二次側のLOCKストラクチャー内に“固定(hardened)”。
 - CF上のLOCKストラクチャーとSCAストラクチャーを二重化しながら、多拠点Sysplex運用における実用的な距離を拡大。
- LOCKストラクチャーの非同期二重化の前提
 - DB2 12でAPAR PI66689を適用
 - IRLM 2.3 APAR PI68378
 - z/OS V2.2でAPAR 0A47796
 - CFCC CFLEVEL 21、サービス・レベル02.16以上(z13以降)

CF LOCKストラクチャー二重化 - 従来の動き



CF LOCKストラクチャー二重化 - 非同期二重化の動き



DB2 12の主要フィーチャー



システム・パフォーマンス

- メモリー最適化の活用
- 連続したBPOOL
- INSERTアルゴリズム
- コミットLRSN処理改善
- zIIPサポートの拡大
- スケーラビリティの向上
- DRDA LOAD
- 静的SQLのためのRLF



SQL パフォーマンス

- 動的プラン固定化
- RUNSTATS
プロファイル
- UNION ALL、OUTER
JOIN照会でのCPU向上
- SORTの向上
- 適応索引
- UDFキャッシュ



アナリティクス、クラウド、モバイル

- IDAA
- RestAPI
- ページネーション
- 断片ごとの削除
- XM、JSON処理の向上
- 複雑なトリガー
- テンポラル関連拡張
- 拡張MERGE
- SQLPLの向上



DBA、365日24時間の可用性

- 簡易な移行
- オンライン・スキーマ
- PBR RPN
- セキュリティー
- 非同期LOCK二重化
- LOBのzEDC圧縮
- プロファイル利用の向上
- テンポラルRTS
- ユーティリティーの向上

Thank You

Akiko Hoshikawa

Senior Technical Staff Member
akiko@us.ibm.com

ワークショップ、セッション、および資料は、IBMまたはセッション発表者によって準備され、それぞれ独自の見解を反映したものです。それらは情報提供の目的のみで提供されており、いかなる参加者に対しても法的またはその他の指導や助言を意図したのではなく、またそのような結果を生むものでもありません。本講演資料に含まれている情報については、完全性と正確性を期するよう努力しましたが、「現状のまま」提供され、明示または暗示にかかわらずいかなる保証も伴わないものとします。本講演資料またはその他の資料の使用によって、あるいはその他の関連によって、いかなる損害が生じた場合も、IBMは責任を負わないものとします。本講演資料に含まれている内容は、IBMまたはそのサプライヤーやライセンス交付者からいかなる保証または表明を引きだすことを意図したものでなく、IBMソフトウェアの使用を規定する適用ライセンス契約の条項を変更することを意図したものでなく、またそのような結果を生むものでもありません。

本講演資料でIBM製品、プログラム、またはサービスに言及していても、IBMが営業活動を行っているすべての国でそれらが使用可能であることを暗示するものではありません。本講演資料で言及している製品リリース日付や製品機能は、市場機会またはその他の要因に基づいてIBM独自の決定権をもっていつでも変更できるものとし、いかなる方法においても将来の製品または機能が使用可能になると確約することを意図したものではありません。本講演資料に含まれている内容は、参加者が開始する活動によって特定の販売、売上高の向上、またはその他の結果が生じると述べる、または暗示することを意図したものでなく、またそのような結果を生むものでもありません。パフォーマンスは、管理された環境において標準的なIBMベンチマークを使用した測定と予測に基づいています。ユーザーが経験する実際のスループットやパフォーマンスは、ユーザーのジョブ・ストリームにおけるマルチプログラミングの量、入出力構成、ストレージ構成、および処理されるワークロードなどの考慮事項を含む、数多くの要因に応じて変化します。したがって、個々のユーザーがここで述べられているものと同様の結果を得られると確約するものではありません。

記述されているすべてのお客様事例は、それらのお客様がどのようにIBM製品を使用したか、またそれらのお客様が達成した結果の実例として示されたものです。実際の環境コストおよびパフォーマンス特性は、お客様ごとに異なる場合があります。

IBM、IBM ロゴ、ibm.com、DB2、WebSphere、z13、zEnterprise および z/OSは、世界の多くの国で登録されたInternational Business Machines Corporationの商標です。他の製品名およびサービス名等は、それぞれIBMまたは各社の商標である場合があります。現時点での IBM の商標リストについては、www.ibm.com/legal/copytrade.shtmlをご覧ください。

Adobe、Adobe ロゴ、PostScript、PostScript ロゴは、Adobe Systems Incorporatedの米国およびその他の国における登録商標または商標です。

IT Infrastructure LibraryはAXELOS Limitedの登録商標です。

インテル、Intel、Intel ロゴ、Intel Inside、Intel Inside ロゴ、Centrino、Intel Centrino ロゴ、Celeron、Xeon、Intel SpeedStep、Itanium、およびPentium は Intel Corporationまたは子会社の米国およびその他の国における商標または登録商標です。

Linuxは、Linus Torvaldsの米国およびその他の国における登録商標です。

PowerLinux is a trademark of International Business Machines Corp. The registered trademark Linux is used pursuant to a sublicense from LMI, the exclusive licensee of Linus Torvalds, owner of the mark on a world-wide basis.

Microsoft、Windows、Windows NT および Windows ロゴは Microsoft Corporationの米国およびその他の国における商標です。

ITILはAXELOS Limitedの登録商標です。

UNIXはThe Open Groupの米国およびその他の国における登録商標です。

Cell Broadband Engineは、Sony Computer Entertainment, Inc.の米国およびその他の国における商標であり、同社の許諾を受けて使用しています。

JavaおよびすべてのJava関連の商標およびロゴは Oracleやその関連会社の米国およびその他の国における商標または登録商標です。

Linear Tape-Open, LTO, LTO ロゴ, UltriumおよびUltrium ロゴは、HP、IBM Corp.およびQuantumの米国およびその他の国における商標です。