

DB2アプリケーションパフォーマンス改善の施策 ーパフォーマンス障害ゼロへの挑戦ー



製造・流通システム第2開発部
ITスペシャリスト

高田 直明

Naoaki Takata

リレーショナルデータベースとしてDB2は広く採用されている。その理由の一つに挙げられる点は、RDBMSはSQL言語が手軽に使用でき、アプリケーション開発とその維持管理を容易にするソフトウェアだということである。このSQL機能を実現するために、DB2は自分自身でデータのアクセス経路を決めている。しかし、この仕組みがDB2ユーザからは適切なパフォーマンスが得られているのか分かりにくくしており、問題となっている。

本論文では、開発から維持管理までの全般にわたって、筆者が長年、DB2アプリケーションパフォーマンス改善で実践した解析手順や方法を解説し、SQL解析ツールの開発と活用事例を報告する。

1. はじめに

DB 2 は、今日、メインフレームのリレーショナルデータベース（RDB）として世界中で広く採用されている。この DB 2 を使用したアプリケーションは、プログラム内の SQL 文（以下 SQL と称する）の記述方法や表の索引構成により、パフォーマンス性能が大きく左右される。

アプリケーション開発時は、アプリケーション仕様（性能）を実現するために、SQL や索引をチューニングする必要がある。特に多くの要員で推進する大規模アプリケーション開発では、開発スキルのバラツキが顕在化するために、チューニングが不可欠である（図 1）。

維持管理時では、開発時のチューニングが不十分であると、CPU や DASD（Direct Access Storage Device）など資源が有効活用されないという問題が生じる。また、業務プロセスが変わるに伴い、プログラムやデータベースが変更され、稼動中システムのパフォーマンスも変化する。さらに、リソース面の TCO 削減の観点からも、診断や改善が求められている。

本論文は、開発局面でのパフォーマンス障害ゼロへ向けての対策と、稼動中システムの DB 2 アプリケーションのパフォーマンス改善方法について、筆者が長年経験したこ

とをまとめたものである。内容は、チューニングの手順や手法、開発した SQL 解析ツールの活用方法などについて紹介している。

なお、本論文で提案したパフォーマンス改善の施策は、DB 2 に限定されるものでなく、RDBMS であれば、その内部構造を解析することによって適用可能なものである。

2. パフォーマンス上の問題点と原因

DB 2 のデータをアクセスする経路（アクセスパス）は、DB 2 が持つ複雑な内部ルールによって、オプティマイザが、DB 2 の統計情報（DB 2 カタログ）の中身を参照して決定する。このアクセスパスは、SQL を基にコスト計算（CPU、I/O）が行われ、一番コストの少ないものが決定される。なお、同じ SQL でも、DB 2 の統計情報の値により、コスト計算結果が異なり、アクセスパスが違って来る事がある。また、このアクセスパスは種類が多く、それぞれ特徴を持ち、パフォーマンスに大きな差が出てくる（図 2）。

このアクセスパスが与える問題点と原因を、開発局面と維持管理局面から説明する。

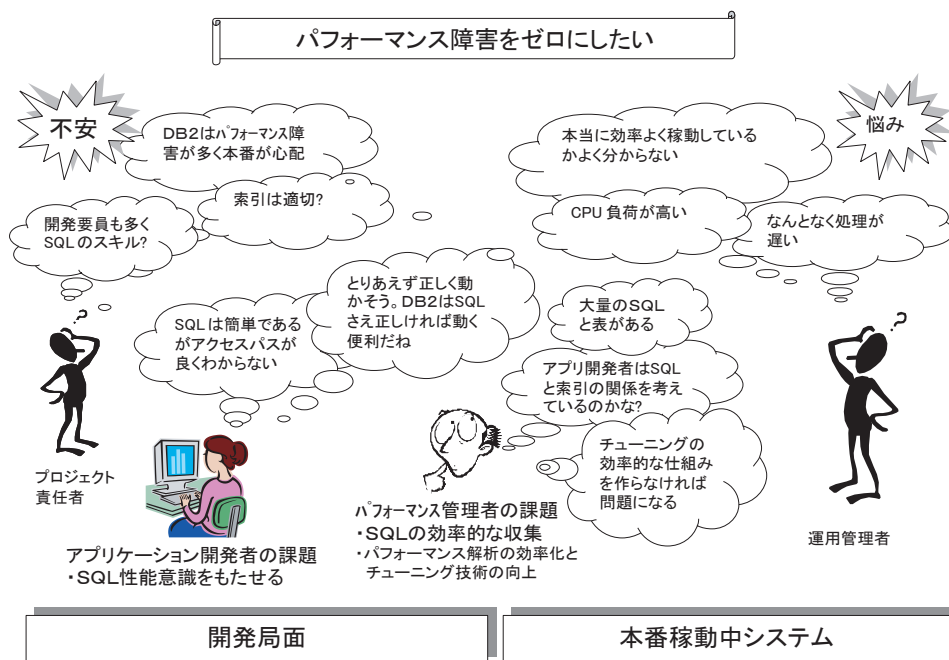


図1 DB 2 パフォーマンスの問題点と課題

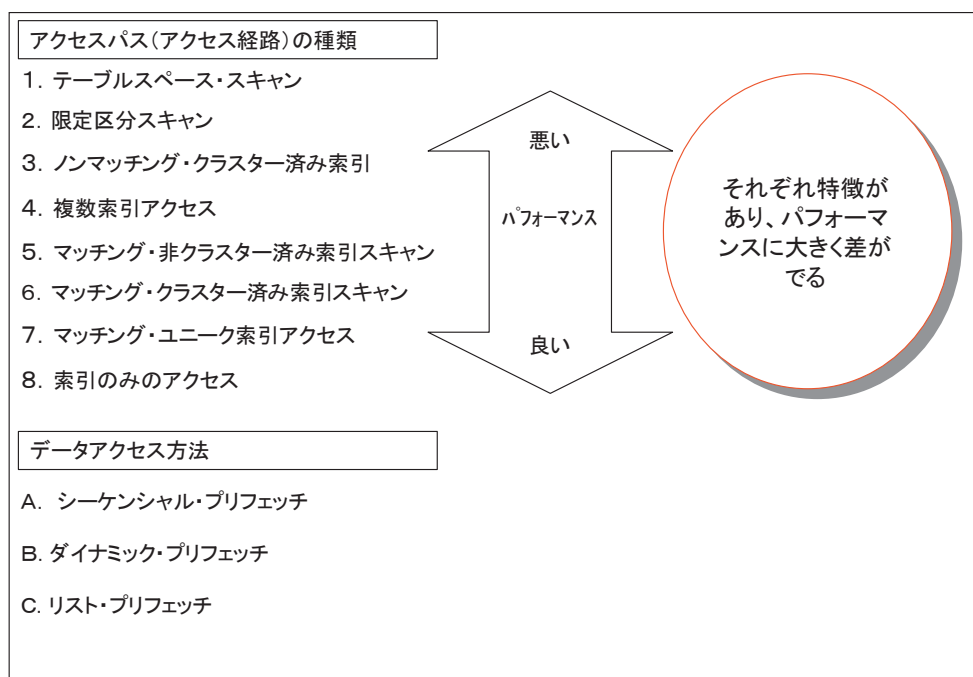


図2 アクセパス（アクセス経路）の種類とデータアクセス方法

2.1. 開発局面での問題点と原因

SQLと索引の検証が確実に実施されず、本番サービスイン直後に、パフォーマンス障害の問題が発覚することが少なくない。その原因を以下に述べる。

①パフォーマンスを意識した開発になりにくい

SQLは文法さえ正しければ、パフォーマンスに影響を与える索引やSQLのWHERE述部を意識しなくても、データを処理する。この点が、RDBを使用したアプリケーションの開発を容易にしている一方、アプリケーション開発者の多くは、パフォーマンスを意識せずに開発する結果となっている。

②SQLの変化に対応できず、最適な索引設計ができていない

SQLやテーブル・索引は、開発規模が大きくなるほど、追加／変更が頻繁に発生する。開発システム全体を考慮した最適索引設計が適宜必要となるが、この変化に対応する手段や方法が確立されていない。

③開発局面のテストでは、SQLと索引が最適設計されているか分からない

開発段階では、性能よりも機能の実現が優先されるため、少ないデータでテストされることが多い。また、本番に近いデータ量が準備できないことも多い。このようなことか

ら、開発局面でのパフォーマンス検証として、システム負荷テストが行われる。しかし、主要な処理やパフォーマンスに関する重点処理についてのみ、性能問題を解決することが多い。そのため、アプリケーション全体から見て、最適なSQLの実行と索引設計がされているか否かが問題として残る。

2.2. 維持管理局面（稼動中システム）での問題点と原因

システムは安定稼動していても、パフォーマンス面での適正稼動という点では疑問が残ることがある。この原因について以下に述べる。

①開発局面のパフォーマンス問題未解決で本番稼動

本番サービスイン後、初期障害を解決したシステムは安定稼動する。しかし、パフォーマンスが安定してきても、SQLは適正稼動しているのかと疑問を感じている運用管理者は多い（図1）。なぜなら、開発局面で残った効率の悪いSQLや索引が、高いハードウェア性能の陰に隠れていることがよくあるからである。

②業務プロセスの変化によるパフォーマンスへの影響

稼動中のシステムが、アプリケーション業務プロセスにより変更された場合、パフォーマンス面で適正稼動してい

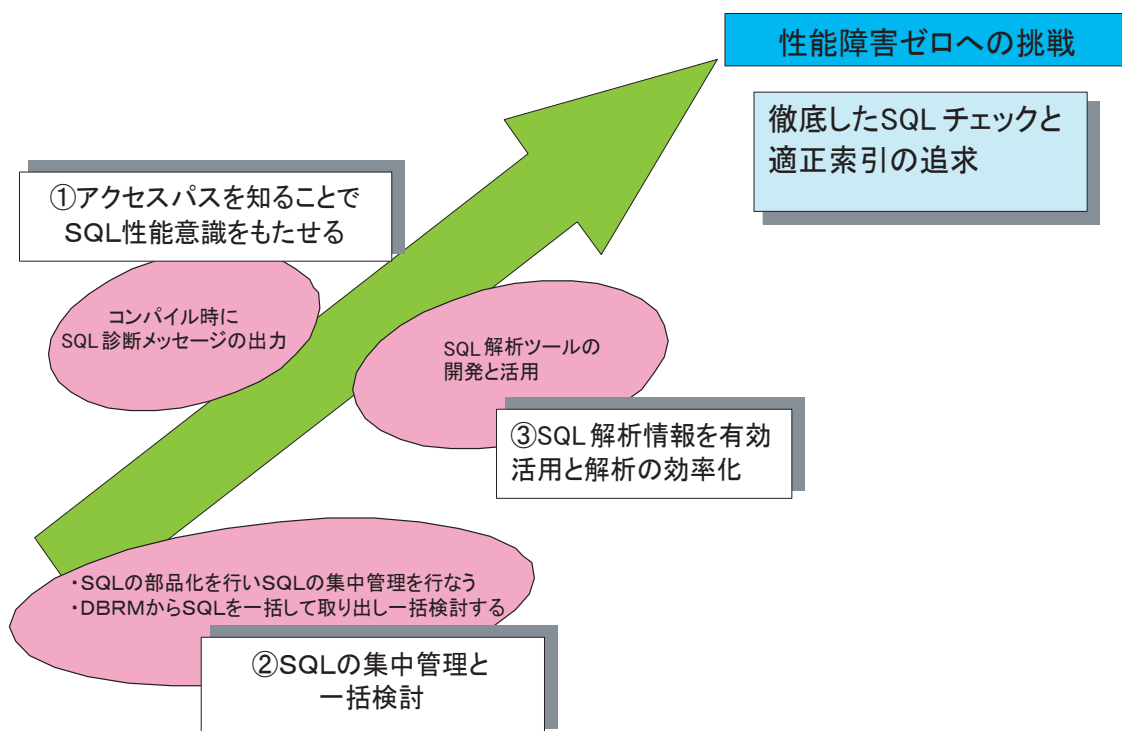


図3 問題解決のための3つの施策

るのか診断が必要である。

稼動中のアプリケーションをチューニングした経験から、不要なデータページアクセスやCPUの無駄遣いが発生し、コンピュータ資源が有効活用されていないケースが、大規模システムになればなるほど多い。このような稼動状態でも、ハードウェア性能向上により、パフォーマンス問題が表面化しないのである。

しかし、CPUの効果的な利用やI/Oの削減、処理時間の短縮により、ハードウェアそのものの費用や運用費の低減、ユーザの業務効率を向上させることができる。

3. パフォーマンスの課題と問題解決の施策

開発局面や維持管理局面の問題点と原因を整理すると、アプリケーション開発者とパフォーマンス管理者の両者の立場からみた課題が出てくる（図1）。パフォーマンス向上を行う上で、解決すべき課題は下記の3点である。

①性能のよいSQLを書く意識をもたせる。

プログラム開発時点では機能の検証が中心となり、パフォーマンスの意識が薄い。また、SQLは容易に扱えるがアクセスパスは複雑で開発者の目に触れないため、性能意

識が薄れがちになる。そのため、プログラム開発時点より、アクセスパスを意識し、性能のよいSQL作りができる仕組みが必要である。

②漏れのない索引設計を行うために、チューニングの基となるSQLを効率よく集める。

アクセスパス分析は個々のSQL分析が必要で、そのための作業工数を見込んでおかなければならない。開発局面で頻繁に変更されるSQLを効率よく収集する必要がある。

③チューニング知識や技術を習得し、パフォーマンス解析と改善を効率よく実施する。

チューニングは、アクセスパスの特徴や、アクセスパスが選択される条件、DB 2 システムがもつ内部ルールに精通したチューニングスキルを必要とする。このスキルを効率よく活用する仕組みが必要である。

上記の①はアプリケーション開発者の、②と③はパフォーマンス管理者にとっての課題である。

これらの課題を解決し、「パフォーマンス障害ゼロを実現」するため、次の3つの施策に取り込んだ（図3）。

a. アクセスパスを知らせることで、SQL性能意識をもたせる。

アプリケーションの開発で、アクセスパスは開発者の目に触れない。これを目にみえるような仕組みを作る必要がある。これにより、アクセスパスの存在を知り、SQL性能意識を高める。

b. SQLの集中管理と一括検討をする。

SQLのプログラム上の組込み方法に応じたSQLの収集を行い、漏れのない索引設計をする必要がある。索引設計で重要なことは、同一テーブルを処理する全部のSQLを漏れなく収集して、索引と照合することである。この中で、アクセスパスや索引が選択される条件を明確にすることで、パフォーマンス性能が確保される。

このSQLの収集方法は、SQLをプログラムの中に直接コーディングする場合と、SQL処理を部品化（サブプログラム）する方法では違ってくる。集中管理方式は、SQL部品化などにより、SQLの変更がパフォーマンス管理者から随時確認できる場合に適用する。また、一括検討方式は、一般的な開発でSQLの変更が随時には確認できない場合に適用する。

c. SQL分析情報を有効活用し解析の効率化を図る。

SQLの性能評価をするには、DB2のオプティマイザが使用するDB2統計情報（索引、索引列、テーブル）や、SQLのWHERE述部などの各種情報を収集する必要がある。また、EXPLAIN情報（アクセスパス）も重要な情報となる。これら別々に管理されている情報を統合する、SQL

解析ツールの開発が必要である。このツールをパフォーマンス改善の手段として利用し、パフォーマンス管理者のチューニング知識や経験が最大限に発揮できるようにする。

4. パフォーマンスチューニングの視点と手順

DB2のパフォーマンスに影響を与える要因は、アプリケーション側とシステム側とに分けられる。

アプリケーション側の要因としては、テーブルと索引の論理・物理設計とDB2アプリケーションのSQLがある。また、システム側の要因は、CPU能力、DASD配置やDB2システム環境設定になる。このテーブル・索引の論理設計とSQLは、改善の手がつけやすく、改善効果も大きい。

図4は、DB2パフォーマンス決定要因がアクセスパスに与える影響と問題点をまとめたものである。

本論文では、この図4の（2）テーブル・索引の論理設計と（3）DB2アプリケーション（SQLステートメント）の二点に注目して、チューニングの視点と手順を論述する。

4.1. チューニングの視点

DB2のパフォーマンスは、SQLのWHERE述部と索

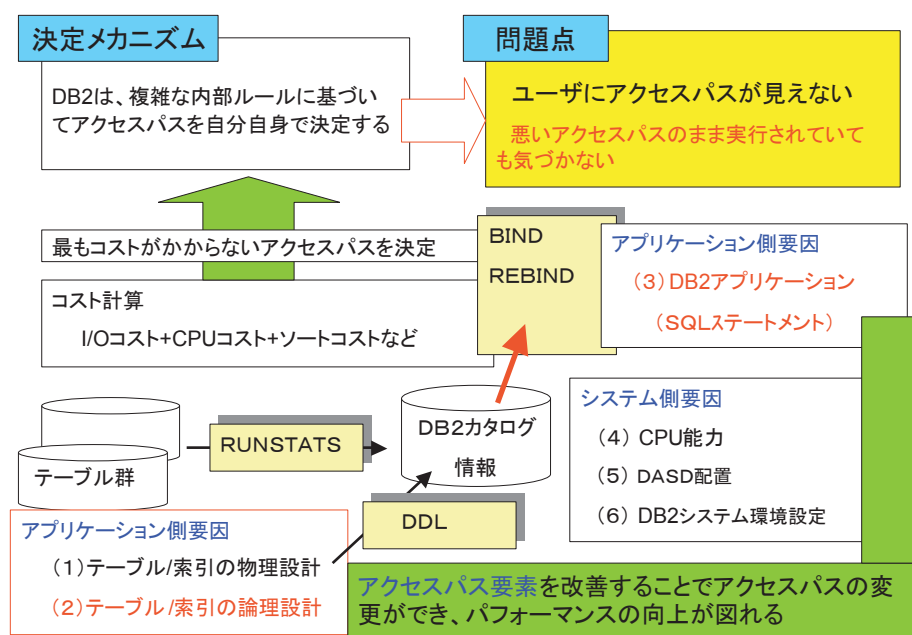


図4 DB2アクセスパスの決定メカニズムと問題点

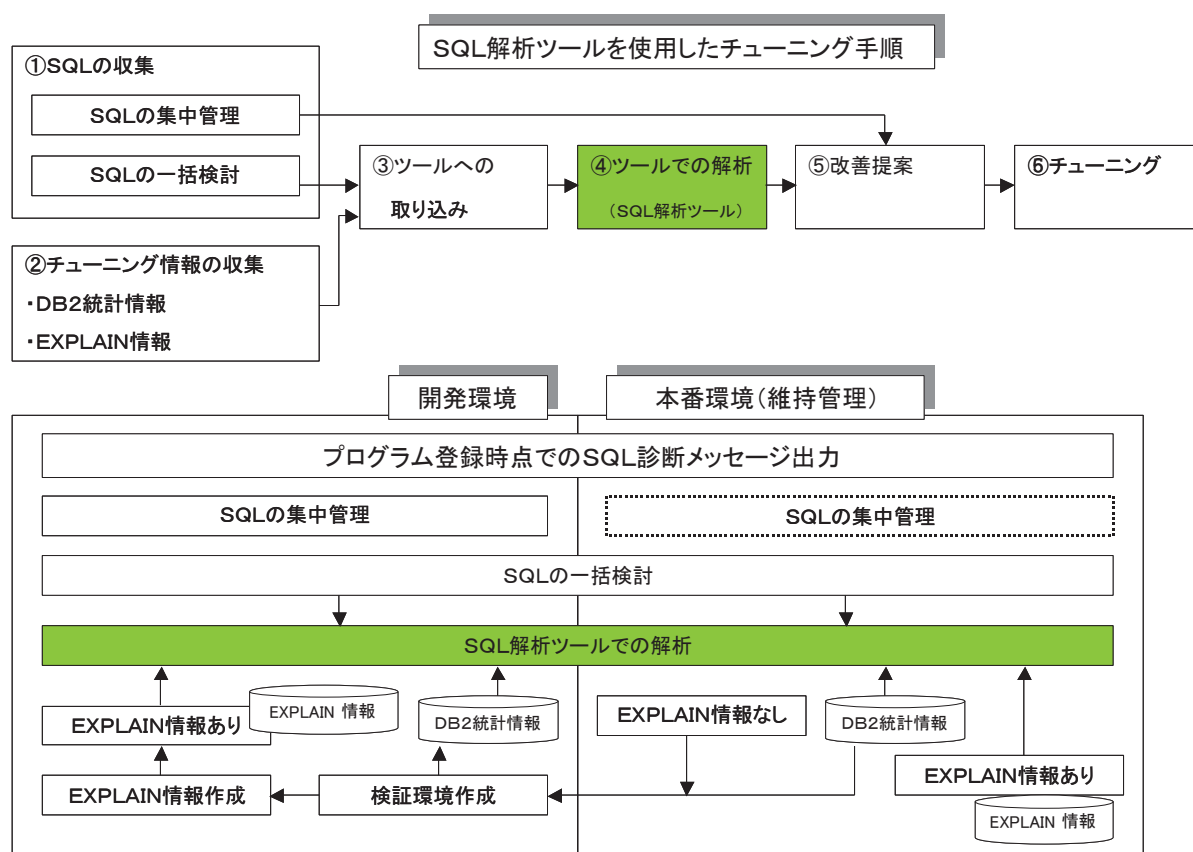


図5 DB 2 アプリケーションパフォーマンス解析手順

引構成によりアクセスパスが決まり、大きく差が出る。

WHERE述部に対する索引がないと、テーブル全体のデータを読み込む。また、索引列の最初の列を使用する述部がないと索引を全て読み込む。索引への列追加や入れ替えにより、同一アクセスパスでもデータページの読み込み件数やCPU負荷が違ってくる。SQLにソートの指定(ORDER BY)があっても、索引の構成によりソートが発生しないなど、チューニング要素が多く、奥が深いのである。

同じエンジンを載せた車で目的地に行くのに、高速道路、一般道路、山道では到着時間が変わってくる。また、同じ山道を一人のドライバーで完走するのか、二人、三人の複数で完走するのか、メンバや走順により差が出る。このエンジンがSQLで、高速道路や山道がアクセスパスである。ドライバー(人)が索引の列で、人や人の組合せ(列順)が索引になる。SQLと索引のチューニングとは、早く目的地に到着するため、エンジンの性能を向上し、最強のメンバと走順を決め、最適な道を選択することである。

DB 2 がアクセスパスを決めるのではなく、人がアクセスパスとアクセス方法を決めてDB 2 に使わせるのである。これが、チューニングの視点である。

4.2. チューニング手順

アプリケーション全体のSQLをアクセスパス分析してチューニングを行うのは、確立した手順やツール無しでは難しい。その原因は、SQLやテーブル、索引の数の多さ、SQLの頻繁な変更、チューニングに必要なデータの分散にある。

図5は、アプリケーション開発プロジェクトや本番稼動中のシステムで実施してきたチューニング手順の概要で、後述する施策や実現方法が手順としてどこに位置付けられているかを示したものである。

5. 実現方法

5.1. プログラム登録時にSQL診断メッセージを出力

効率の悪いSQLが記述された場合、プログラム登録時点でSQL警告メッセージを出力して性能に意識を向ける。EXPLAINを利用して、テーブルスペーススキャンや、ノ

ンマッチング索引スキャンなど、パフォーマンスに影響を与えるSQLは警告メッセージを出力する。また、使用されているアクセスパスや索引のマッチカラム度を表示することで、索引によるデータの絞込み件数が想定できるようになり性能を意識したプログラムを作成することができる(図6)。

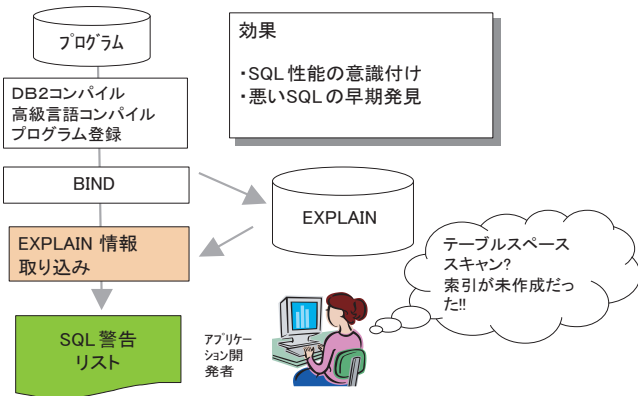


図6 プログラム登録時点でSQL診断メッセージを出力

5.2. SQLの集中管理と一括検討

5.2.1. 部品化(サブプログラム化)によるSQLの集中管理

SQLを部品化(サブプログラム、以下、IO部品と称する)して、SQL単位に登録する仕組みを作り、SQL集中管理を実現する(図7)。プログラム(SQL)の実行前に、パフォーマンス管理者の元へSQLが集まり、集中管理が可能になる。この結果、個々のSQLの性能チェックと索引検討が、SQLを変更するたびに確実に実現できるようになる。

このIO部品は、エクサ社内で蓄積された技術を基に、数社に適用してきたDBI (Data Base Interface) の方式である。

SQLの集中管理と解析方法は次のようになる。

- ① SQLの収集
- ② SQLの検証とテーブル別SQL情報表(表1)の作成
- ③ 登録されている索引の性能面からの検証

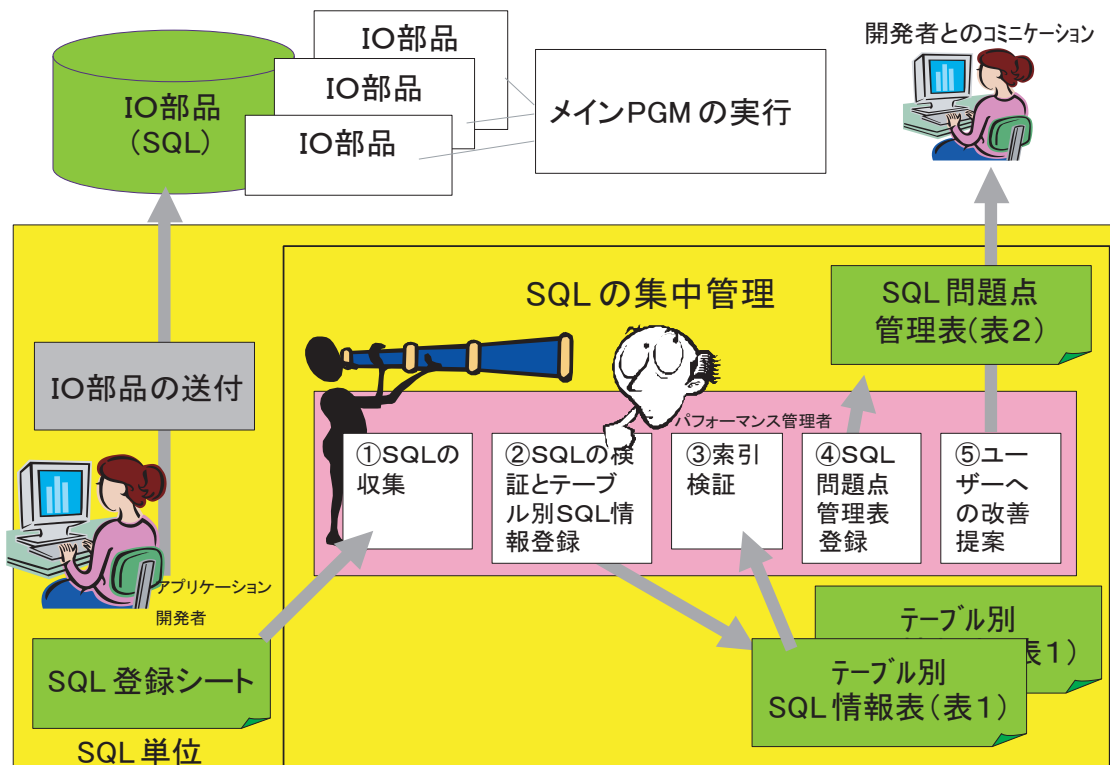


図7 IO部品作成とSQL集中管理の流れ

表1 テーブル別SQL情報表

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD
1																														
2																														
3																														
4																														
5																														
6																														
7																														
8																														
9																														
10																														
11																														
12																														
13																														
14																														
15																														

表2 SQL問題点管理表

	A	B	C	D	E	F	G	H	I	J
1	問題番号	登録申請日	検証日	アクセス/ア	申請者	問題点・不明点	問題有	問題無	回答者	回答
2	83	4月18日	4月19日	CCIDX5W	江草	4/19(高田)索引3が該当しますが、索引3の列に 依頼先表コードと予定データ属性を追加すれば、 索引3だけでデータの絞り込みができ、パフォーマンス がよくなります。索引3の変更検討をお願いします。	○		江草	4/22 索引3の変更をいたし
3	64	4月18日	4月19日	CCIDX5X	江草	問題番号83と同じ	○		江草	4/22 索引3の変更をいたし
4										

- ④問題点や改善提案をSQL問題点管理表(表2)にする
 ⑤問題点管理表による開発者とのコミュニケーションによるパフォーマンスチューニングの提案

以下、ここで使用されるテーブル別SQL情報表(表1)と、SQL問題点管理表(表2)について記述する。

①テーブル別SQL情報表(表1)

DB 2のテーブルをアクセスする全てのSQLのWHERE

述部を集めた表である。SQLの登録や変更時に、定義済みの索引と比較することで、索引の有無や索引のマッチカラム度(述部が索引にマッチする数)が分かり、パフォーマンスの悪いSQLの発見や最適な索引の検討ができる。

SQLコーディング規準に違反したSQLを早期に発見することも可能となる。

②SQL問題点管理表(表2)

上記の登録・修正時に、SQLや索引に問題があれば改

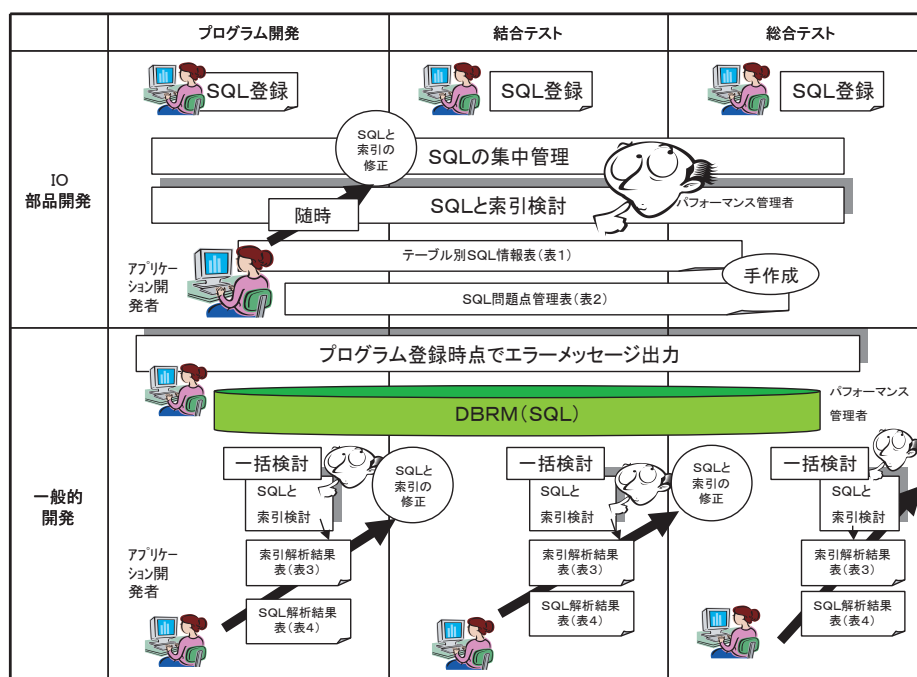


図8 部品開発と一般的開発でのSQLと索引検証方法

表3 索引解析結果

[illegible]

表 4. SQL解析結果

Microsoft Excel - 編集内容:alt

ファイル(F) 編集(E) 表示(V) 挿入(I) 書式(O) ツール(T) データ(D) ウェブ(W) ヘルプ(H)

K13 集計 1/3

	A	E	G	D	E	F	G	H	I	J	K
1	SQL 解析結果										
2	テーブル名: TY1601										
3	番号	調査対象	PGM名+番号	索引の追加	索引の追加	効果の追加	索引の追加	ソートの回避	評価	索引名	マッチラム
4	1	○	Y16519B 529	新2	2/4	○	○	○	○	Y160101	0
5	2	○	Y16518B 529	新2	2/4	○	○	○	○	Y160101	0
6	3	○	Y1651EB 527	新2	2/4	○	○	○	○	Y160101	0
7	4	○	Y16525B 524	新2	2/4	○	○	○	○	Y160101	0
8	5	○	Y1651EB 521	新2	2/4	○	○	○	○	Y160101	0
9	6		Y16523B 695							Y160101	3
10	7		Y16523B 937							Y160101	3
11	8		Y16523B 683							Y160101	3
12	9		Y16523B 907							Y160101	3
13	10		Y16509B 553	新3	2/4	○		▲		Y160101	1/3
14	11		Y16517B 536	新3	2/4	○		▲		Y160101	1/3
15	12		Y16501B 553	新3	2/4	○		▲		Y160101	1/3
16	13		Y16510R 554							Y160101	1
17	14		Y16524B 524							Y160101	1
18	15		Y1651DB 539							Y160101	1
19	16		Y1650EB 539	新2	1/4	○	○	○	○	Y160101	0

善案としてこの表にまとめ、アプリケーション開発者に提示してアプリケーション開発者がチューニングを実施する。

るツールである。

5.2.2. DBRMよりSQLを抽出し一括検討

S Q Lの変更が随時に把握できない開発では、開発局面でチューニングを必要とするタイミングでS Q Lと索引の検討を一括して実施する(図8)。この時点で、D B R M (DataBase Request Modules)に登録されている全S Q Lと、アクセス分析に必要なデータを同期してツールに取り込む。取り込んだデータを基に、ツールを使用してS Q Lと索引のアクセスパス分析を行い、改善提案を短期間に実施する。

検討結果と改善内容を、索引解析結果表（表3）とSQL解析結果表（表4）に示す。これらの表にある解析元データは、ツールで自動生成される。これを基にSQLと索引の改善提案内容を、同表に反映したものである。

5.3. SQL解析ツールの開発と活用

われわれが開発したSQL解析ツールは、SQLのアクセスパスを分析し、SQLや索引の問題点と対策を提示す

5.3.1. SQL解析ツールの特徴

①EXPLAIN情報、DB 2カタログ情報、SQLの情報を有効活用

- これらの情報をツールに取り込み、アクセスパス分析に活用。
- S Q LをEXPLAIN情報と結合して、分析結果の理解を容易化。

②アクセスパス上の問題点抽出と解決方法の提示

- ・アクセスパスの分析に基づいて、問題の具体的な解決方法（対策）を提示。

③使い易いツール

- ・視覚的に分かり易い、パソコン上でのEXCEL機能を活用したツール
- ・EXPLAIN情報の分析結果を日本語で分かりやすく表示

④索引とSQLの関連を的確に解析

- パフォーマンスの悪いSQLの検索条件と索引定義を解析することで、索引の最適化を支援

⑤関連するアプリケーションの横断的分析を実現

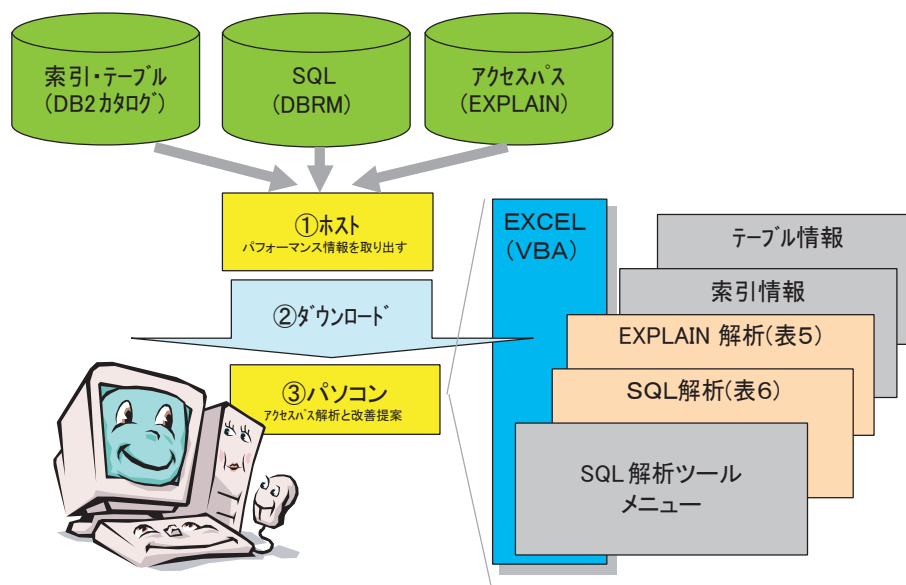


図9 パフォーマンス情報の取り込みとツールの活用手順

- ・あるテーブルに関連する全てのアプリケーション（プラン、パッケージ）のSQLを串刺しで分析
- ・チューニングすることで影響の出るアプリケーションのSQLを容易に把握

- ・索引情報
- ・テーブル情報

5.3.2. 各種情報の抽出方法とツール活用手順

SQL解析ツールの活用手順を以下に示す（図9）。

- ①メインフレーム上のDB 2統計情報（索引・索引列・テーブル）、DBRM情報（SQL）やEXPLAIN情報を取り出す。
- ②抽出したデータをパソコンへダウンロードする。
- ③SQL解析ツールで、EXPLAINの各データに対応するSQLのWHERE述部を付加し、EXPLAINの内容を日本語で分かりやすく表示する。このEXPLAIN情報を基に、パソコンへ取り込んだパフォーマンス解析データを利用し、SQLと索引の解析を行って、アクセスパスの表示や問題点の指摘と対策を提示する。このSQL解析ツールは、EXCELのVBA(Visual Basic for Applications)で作成している。

ツール上での画面（シート）の主な内容は以下である。

- ・SQL解析メニュー
- ・EXPLAIN解析結果（表5）
- ・SQL解析結果（表6）

5.3.3. SQL解析ツールでの解析結果

EXPLAIN解析結果（表5）とSQL解析結果（表6）は、ツール上で関連性を持たせている。EXPLAIN解析結果の行番号を選択することで、SQLごとにアクセスパスの分析を行い、SQL解析の結果を作成する。

この表は、相互に遷移できることが特徴となっている。

①EXPLAIN解析結果（表5）

EXPLAINには、コードや数値が含まれる。また、値の組合せで意味を持つ物もある。例えば、アクセスパスの種類は、アクセスタイプ、マッチングカラム、インデックスオンリーの値の組合せで決まり、加工前のデータは理解しにくい内容である。この解析結果は、EXPLAINデータを解析し、日本語で分かりやすく表示することによりチューニングへ活用するものである。ここに、SQLの分析に必要なWHERE述部を追加しているため、EXPLAINデータとの関係の把握が容易になり、パフォーマンス改善の生産性が著しく向上する。

この解析結果を使用して、テーブル単位にアクセスしている全てのWHERE述部の抽出がEXCELの機能（フィルタ）で可能となり、最適索引設計が漏れなく実現できる。

②SQL解析リスト（表6）

が特にSQLにも問題がない場合、同一画面に表示されたWHERE述部、索引・テーブル情報やEXPLAIN情報をチェックすることで、なぜ予想と違うのかの推測ができる。また、表示されたデータを利用することで、保有しているチューニングスキルと組み合わせ、さまざまな角度から迅速に不適性アクセスパスの原因調査と改善が可能となる。

この解析ツールはEXCELで開発されているため、SQLの解析（表5）でテーブル・索引・アクセスパス・プロ

EXPLAIN内容+日本語表示+SQLのWHERE記述 紙面上、一部表示を省略する)

表 6 SQL解析結果[illegible]

グラムなど、解析のカテゴリ単位にデータをフィルタリングして、用途別にSQLをさまざまな角度から抽出することで、要求するSQLのアクセスパス分析が迅速にでき、使いやすいツールである。

表6のSQL解析結果例は、「ノンマッチング索引スキャンの発生」と「RUNSTATSの未実行」とをツールが自動解析し、この二つの問題点とそれぞれへの対策を表示したものである。アクセスパスで使用されている索引の色表示（黄色）や索引列の緑色表示を見れば、WHERE述部と索引の関係を容易に確認できる。さらに、問題の値をもつセルは赤色表示、主要なキーワードはセルにカーソルを当てただけで意味（HELP）がポップアップ表示するなど注意を喚起するよう工夫をしている。そのため、非常に理解しやすく使用方法も簡単のため、熟練技術者でなくても、効率的な問題判別と解析が容易にできる。

6. 稼動中システムのDB 2アクセスパス分析

維持管理局面におけるDB 2アプリケーションのチューニング方法について事例を紹介する。

システム単位やサブシステム単位に、稼動中システムの全てのSQLと関連するDB 2統計情報、EXPLAIN情報を抽出して、総合的にSQLと索引のシステム診断（クリニック）をする方法である。

6.1. 最適索引検討とDBアクセスパス解析ツールの仕組み

基本的な解析は、5.3.2.の各種情報の抽出方法とツール活用手順で紹介した内容と同様である。

ここでは、DB2PM（DB2 Performance Monitor）より抽出した、性能の悪い特定PLANを解析した事例を紹介する。

図10に、本番環境におけるシステムのDB 2アクセスパス分析と入出力情報を示す。

ここでの解析結果の成果物は、索引解析結果（表3）、

表7 PLAN別改善効果表

Microsoft Excel - 論文内容.xls																			
Q27																			
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	PLAN別改善効果表			凡例: 効果小 ▲ ● ○ ◎ ◎◎ 効果大															
2	テーブル名	対応	PGM名	効果	現状	PL1674	PL1675	PL1783	PL2251	PL2471	PL2551	PL2552	PL2771	PL2851	PL3002	PL3553	PL3631	PL3651	PL367
3	1 TY1601	新2	Y1550EB	◎◎	ノンマッチング索引スキャン														
4			Y15518B	◎◎	ノンマッチング索引スキャン														
5			Y15519B	◎◎	ノンマッチング索引スキャン						◎◎								
6			Y1551EB	◎◎	ノンマッチング索引スキャン						◎◎								
7			Y1551EB	◎◎	ノンマッチング索引スキャン										◎◎		◎◎	◎◎	◎◎
8			Y15525B	◎◎	ノンマッチング索引スキャン										◎◎		◎◎		
9		新3	Y15501B	▲			▲												
10			Y15509B	▲			▲		▲										
11			Y15517B	▲							▲		▲						
12	2 TY1605	新3	Y15500B	▲			▲	▲						▲	▲				
13	3 TY1610	新2	Y51201B	○					○										
14	4 TY1611	新5	Y52913B	◎							◎								
15		新6	Y52907B	◎							◎								

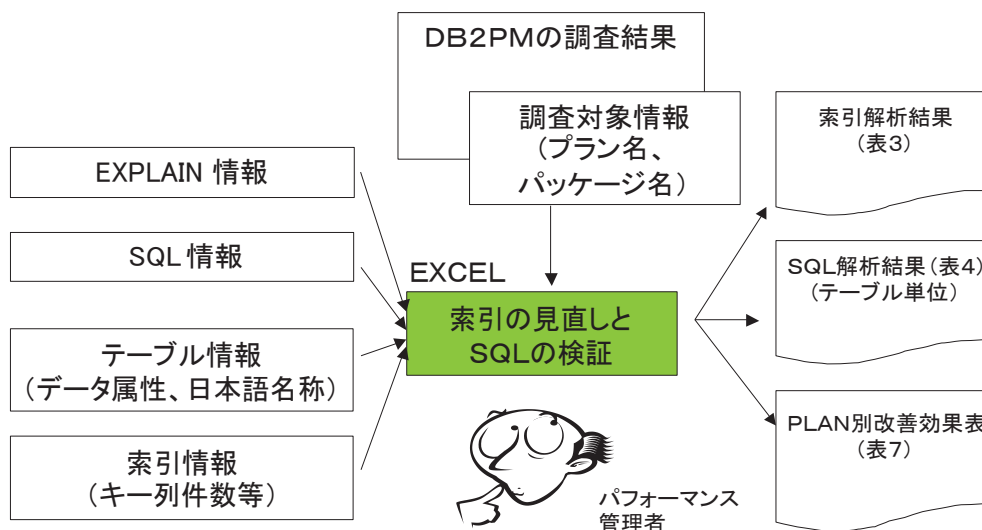


図10 稼動中システムのDB2アクセスパス分析

SQL解析結果(表4)とPLAN別改善効果表(表7)になる。

以下、PLAN別改善効果表(表7)について記述する。

この表は、索引解析結果(表3)とSQL解析結果(表4)をまとめた表で、チューニング後の影響度を知ることができる。

PLANが使用しているプログラムとテーブルや索引がマトリックス表になっている。索引やSQLを改善した場合の効果が、PLAN単位、テーブル単位に串刺しで確認できる。これを利用して、チューニングの優先順位を決定する。

6.2. 稼動中システムでEXPLAIN情報が無い場合の解析

稼動中システムのDB 2アクセスパス分析を実施したいが、EXPLAIN情報が採取されず、現状どのようなアクセスパスを使用しているか不明で困る場合がある。この解決法は、稼動中システム(本番環境)の情報を抽出し、検証用環境に複写してミニ環境を作成することで可能となる(図11)。

①検証用環境作成

DB 2のDDL (Data Definition Language) 生成のための情報を、DB 2カタログ情報から抽出し、パソコン

にダウンロードする。その後、パソコンで検証環境用のDDLを生成して、メインフレームへアップロードして検証用のミニ環境を構築する。検証環境用のデータベースのスペース容量は、データの格納の必要がないため、最小の単位で作成する。

②DB 2カタログ情報の抽出と取り込み

検証環境でアクセスパス分析をするには、本番環境のDB 2カタログ情報を検証環境に反映する必要がある。

③BINDを実行しEXPLAIN情報を作成

本番環境のDBRMを検証環境に複写し、BIND (EXPLAIN (YES)) することで、EXPLAIN情報を生成する。

この手順により、アプリケーションで使用する実データが無くても、DB 2のアクセスパスの解析が実施できる。

7. 適用効果と今後の課題

DB 2アプリケーションの性能向上の施策を数例のプロジェクトに適用した。その効果をまとめると、次の3点になる。

①本番サービスイン時のDB 2パフォーマンス障害ゼロの達成。

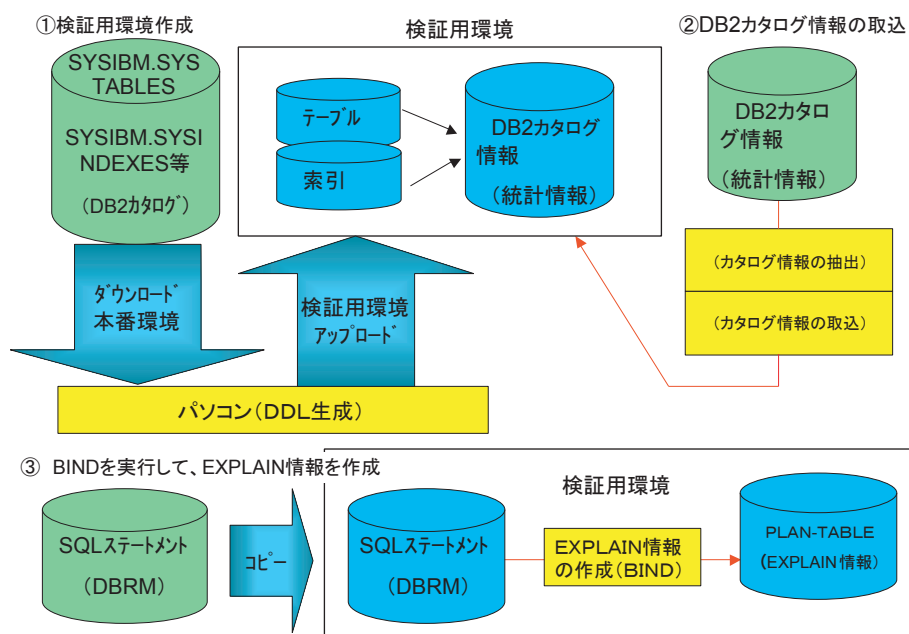


図11 本番環境でEXPLAINデータがない場合の解析

- ②SQLと索引の問題点発見、および改善提案が短期間で可能。
- ③アプリケーション開発者の性能意識が改善。

今後の課題としては、ツール群の機能向上と市販ツールとの連携による利用範囲の拡大がある。

①開発ツールの機能向上

SQLのJOIN・UNION・副照会・動的SQLなどがサポートできていない。また、自動解析結果の問題点指摘と対策も充実する必要がある。

②市販ツールとの連携による利用範囲の拡大

最近、DB 2 アプリケーションパフォーマンス管理ツールが市販されるようになってきた。これらの市販ツールは、チューニングに有効なデータを提供している。この有効データをユーザ側の観点から活用し、ツールの利用範囲を拡大していく必要がある。

＜問い合わせ先＞

製造・流通システム第2開発部

製造・流通第4チーム

Tel. 044-540-2495 高田 直明

E-MAIL: naoaki-takata@exa-corp.co.jp

8. 終わりに

アプリケーションのパフォーマンス障害は、事前に防止されるべきである。筆者が携わったアプリケーション開発でも、多くの索引やSQLに問題が見つかり、改善を実施してきた。また、本番稼動で適正に稼動しているアプリケーションに対してシステム診断（クリニック）を行った結果からも、業務プロセスの変化などから、改善すべき索引（追加や変更）やSQLが数多く見つかっている。

現在、ハードウェア性能が高くなり、パフォーマンス上の問題が出なくなっても、CPUやDASDなどの資源が無駄使いされていることが少なくない。システム資源の有効活用を考えれば、DB 2 アプリケーションのチューニングは重要である。

今後は、本論文で述べたパフォーマンス改善施策や開発したツールをベースに、DB 2 アプリケーションのための、システムクリニックソリューションとしてまとめていく予定である。

参考文献

- 1) IBM"DB2 for OS/390 適用業務設計 Volume 2" GG88-0077-01