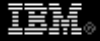



[ホーム](#) | [製品](#) | [サービス & ソリューション](#) | [サポート & ダウンロード](#) | [マイアカウント](#)

[DB2 Developer Domain](#) > [製品別技術情報](#) > [DB2いろはがるた](#) >

DB2いろはがるた



第20回 「ね」 - ネットサーチ、メモリー上をまず探そう



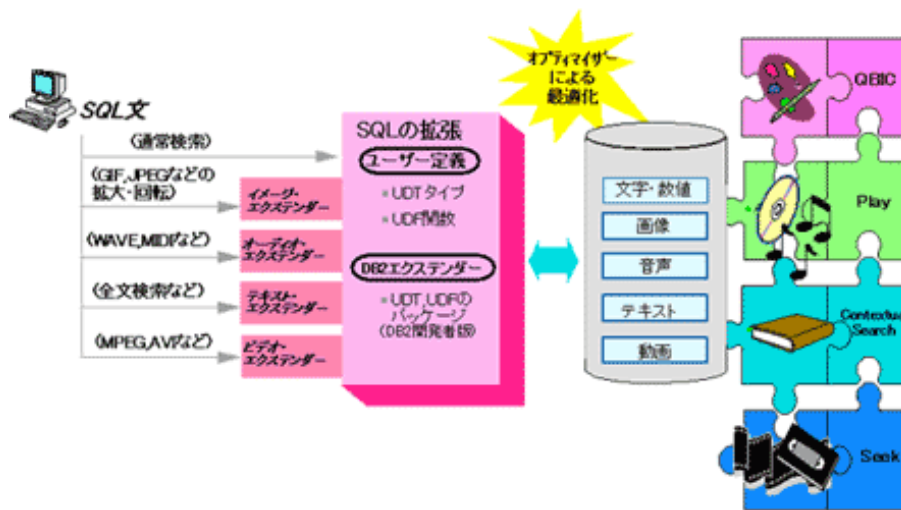
執筆者

春野 さくら

「DB2いろはがるた」を執筆するために参上した、なぞの女性。
日本の古典文学を愛する。

今年の夏の始め頃、プロ野球の世界で8月のキーワードが「リカバリー」になったと聞き、データベースのスペシャリストとしては、「我が意を得たり！」とうれしかったのですが、やっぱり一般受けしなかったらしく、いつのまにか「ミラクル・アゲイン」が主流になってしまいました。やっぱりリカバリーというのは、データベース・スペシャリストだけが、なんとなく居心地のよい特別な言葉なのでしょうか？ とても残念だと思っているのは私だけでしょうか。ともあれ、今回はエクステンダーのお話です。

今日リレーショナル・データベースに保管されるデータの種類としては、従来のテキスト・データだけではなく、音声、イメージ、ビデオ、オブジェクト・リレーショナル型など多種多様なものも主流になってきました。DB2 UDBではそれらのデータを表に保管するために、ユーザー自身が特定のデータ・タイプのためにユーザー定義データ・タイプ(UDT)を作成することができます。一旦そのデータ・タイプが宣言されると、その後はDB2が提供するCHAR型や、INTEGER型と同様に、CREATE TABLE文などで使用することができるし、もちろんSELECTやINSERTも全く他のデータ・タイプと同様に行えます。またそのユーザー定義データ・タイプにあわせて、そのデータを使うSQL関数も作成することができます。これはユーザー定義関数(UDF)と呼び、既存のスカラー関数を元に作ったり、新たにCやJAVA言語を使ってプログラムを作成することも可能です。DB2 UDB製品が提供しているエクステンダーという機能の中味は、実は予めDB2がある特定のデータ・タイプ用に提供するこのユーザー定義データ・タイプとユーザー定義関数の塊のことなのです。もし、ユーザーが使おうと計画しているデータ・タイプが既にこの機能で提供されているなら、このエクステンダーを使えば、即座に、とても簡単に便利に、特殊なデータ・タイプをSQL文で処理することができます。



DB2 UDBが日本語環境で現在提供しているエクステンダーには以下のような種類があります。

テキスト・エクステンダー

長文の文章などを保管し、語句のあいまい検索などを行うことができる。

イメージ・エクステンダー

写真や、イメージなどを保管し、色、形状などで検索することができる

ビデオ・エクステンダー

動画を保存し、シーンの変更などを検索することができる。

オーディオ・エクステンダー

音声を保管し、最初の数秒だけを取り出すことができる。

地理情報エクステンダー

空間情報を保管し、その位置関係、形状などで検索することができる。

XMLエクステンダー

XMLデータを分解して保管したりまた反対に列から組み立てることができる。

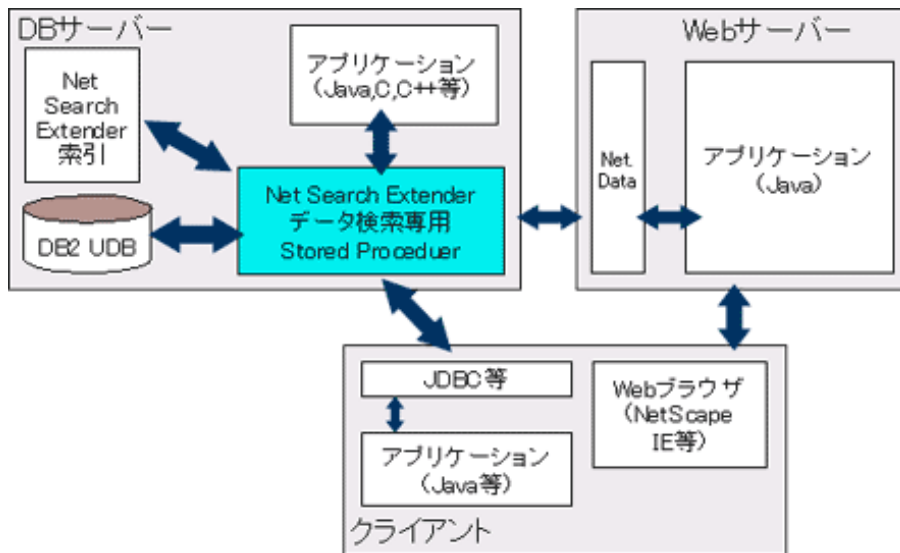
ネットサーチ・エクステンダー

テキストをメモリー展開し、素早く検索することができる

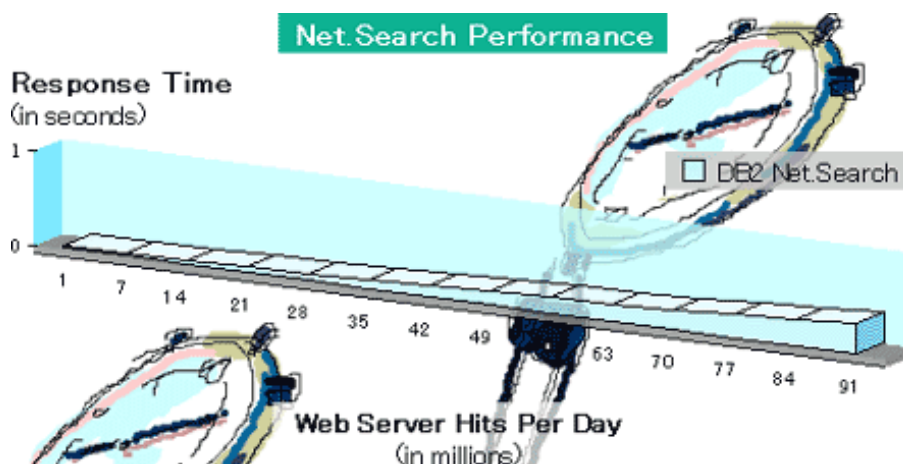
テキスト・インフォメーション・エクステンダー

テキスト・エクステンダーの基本的な機能を実装し、検索エンジンを改良

特に昨年上記のエクステンダーの中で、“業界初のメモリーインデータベース”と大々的にうたったネットサーチ・エクステンダーというフィーチャーがDB2 UDB V7.1で提供されました。



このフィーチャーは、テキスト・エクステンダーというもともとDB2の列に保存されている巨大なテキストを語句検索、ファジー検索する機能をもっている製品を元にして開発されたのですが、大きな違いは表の中の列をあらかじめ、バッファプール外のシェアードメモリー上に展開できるという点です。そのことにより、当然ディスクアクセスより数段上のパフォーマンスが得られるし、またRDBのアクセス機構をバイパスすることによるパフォーマンスの向上も期待できるというわけです。昔から本当に高パフォーマンスが必要なデータはRDBに保管せず、プログラムで予めメモリー上に展開しておくという方法がシステム・デザインでよく採用されていましたが、このネットサーチ・エクステンダーはこのデータのメモリー展開を、DB2 UDBが自動的に行ってくれるのです。ユーザーはRDBのデータと同じようにプログラムではSQL文でアクセスするだけなので、データが本当はどこにあるかは何も意識する必要がありません。どの列をメモリー上に予め展開しておくかの決定は、当然リアルメモリー・サイズの制限とのかねあいがあるので、あらかじめ慎重なキャパシティー計画が必要になります。しかし、Web上の商品カタログのような、大ユーザー数での高いパフォーマンスが必要とされるような業務では大いに威力を発揮すると期待されています。例えば、下のテスト例では9100万ヒット件数のウェブサイト検索負荷テストにおける高パフォーマンス維持を示しています。



今後ハードウェアの進歩に従い、ますます大容量メモリーを搭載したマシンが出現してくるのは間違いないことでしょう。また、今後e-businessにおけるトランザクションの爆発的な増加がおこることは目にみえています。そのような時にこそ、このメモリーイン・データベースの真価が必ず発揮されることでしょう。

[↑ 上に戻る](#)

