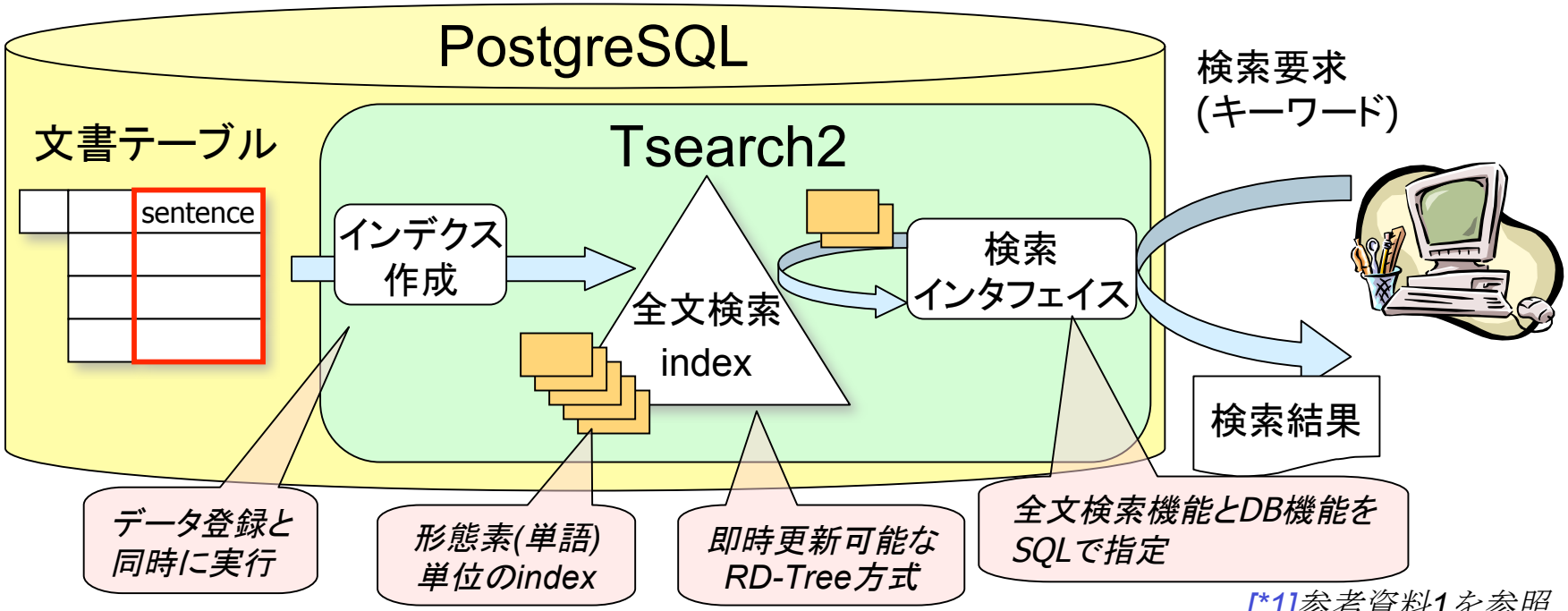
tsearch2日本語化パッチのご紹介

2005年12月2日

NTTサイバースペース研究所
寺本　純司

tsearch2の概要

- PostgreSQLに含まれている欧文向け全文検索モジュール
- 形態素解析方式[*1]による全文検索が可能
  →欧文向けのため、スペース区切りによる索引語抽出
- 即時更新に強いインデクス方式(RD-Tree[*2])が特徴



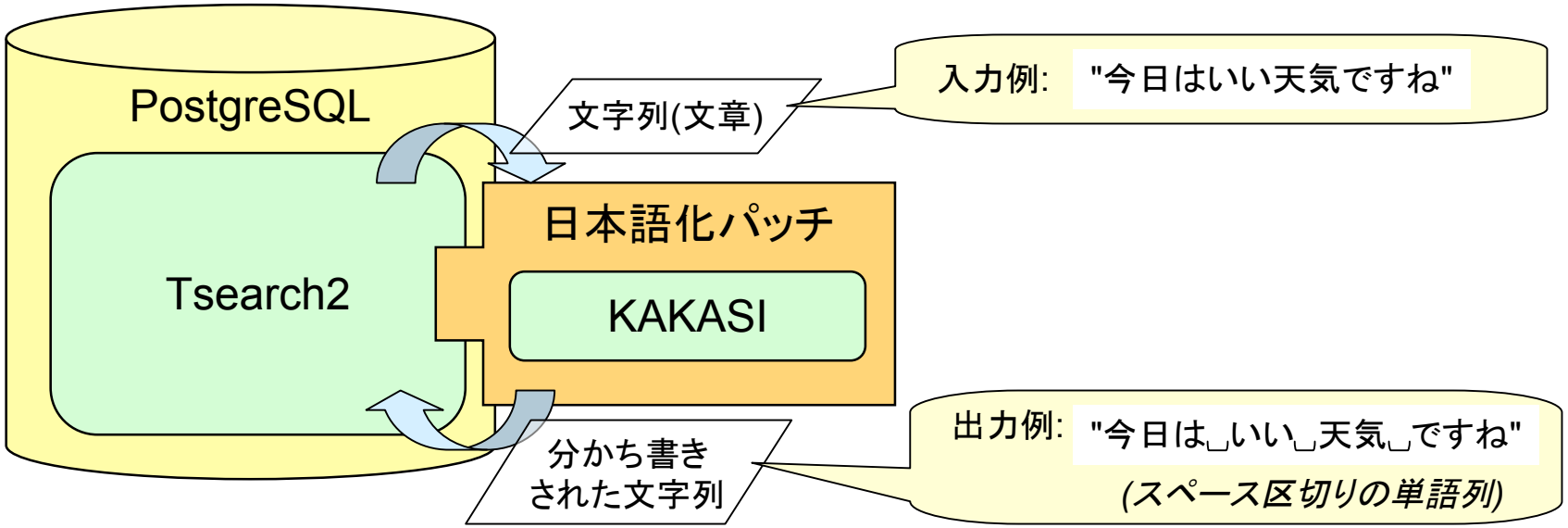
[*1]参考資料1を参照

[*2]参考資料2を参照

日本語化パッチの概要

PostgreSQL機能と既存ツールを生かした、シンプルな方式実現

- tsearch2の基本方式はそのままに、パッチプログラムとして実現
- 日本語形態素解析を行い、単語ごとにスペース区切り(分かち書き)を行う処理のみ追加
- 形態素解析には、OSSツールである「KAKASI」[*]を利用



[*]全文検索エンジンNamazuを代表に、広く活用されている日本語形態素解析ツール

性能

- tsearch2による日本語検索は、中小規模の領域では性能的には十分実用可能
- ただし、大容量のデータを扱うDBMSの機能としては性能的に不十分

[性能評価概要]

※10万文書を対象

	Tsearch2_jp
検索時間	65～200ms
文書登録時間	200ms

文書登録時間： 文書を登録して検索できるようになるまでの時間

[測定環境]
CPU: Pentium III 866MHz×2
RAM: 1.5GB
OS: Fedora Core release 1

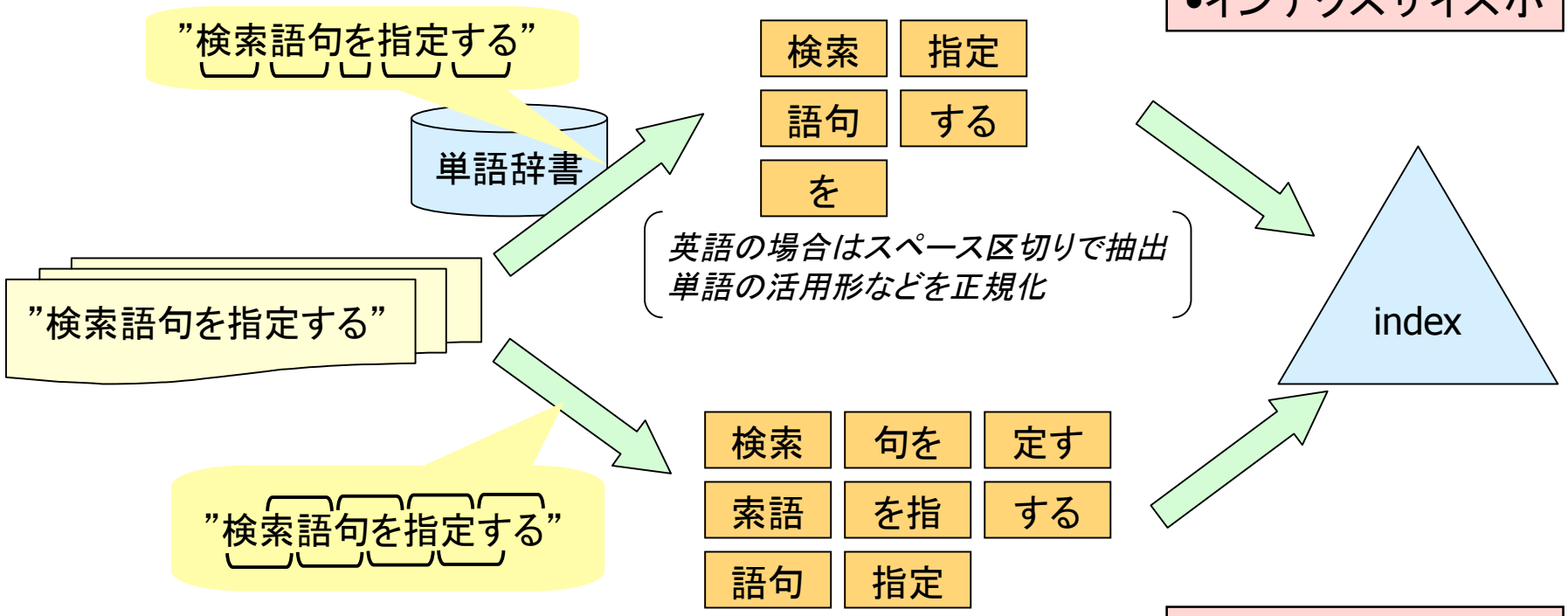
[考察]

- 日本語化Tsearch2に関して、1万文書以上を対象とすると、検索時間にばらつきが出ると同時に、検索漏れも発生することを確認
  (インデクス方式の制限により、単語数が10万語程度までが実用範囲)
- よって、実用可能レベルではあるが、小規模システムに限定される

参考資料1：索引語の抽出方式

(1)形態素解析方式
＝ 形態素(単語)を抽出

特徴：
- 検索漏れがある
- 辞書の保守が重要
- インデクスサイズ小

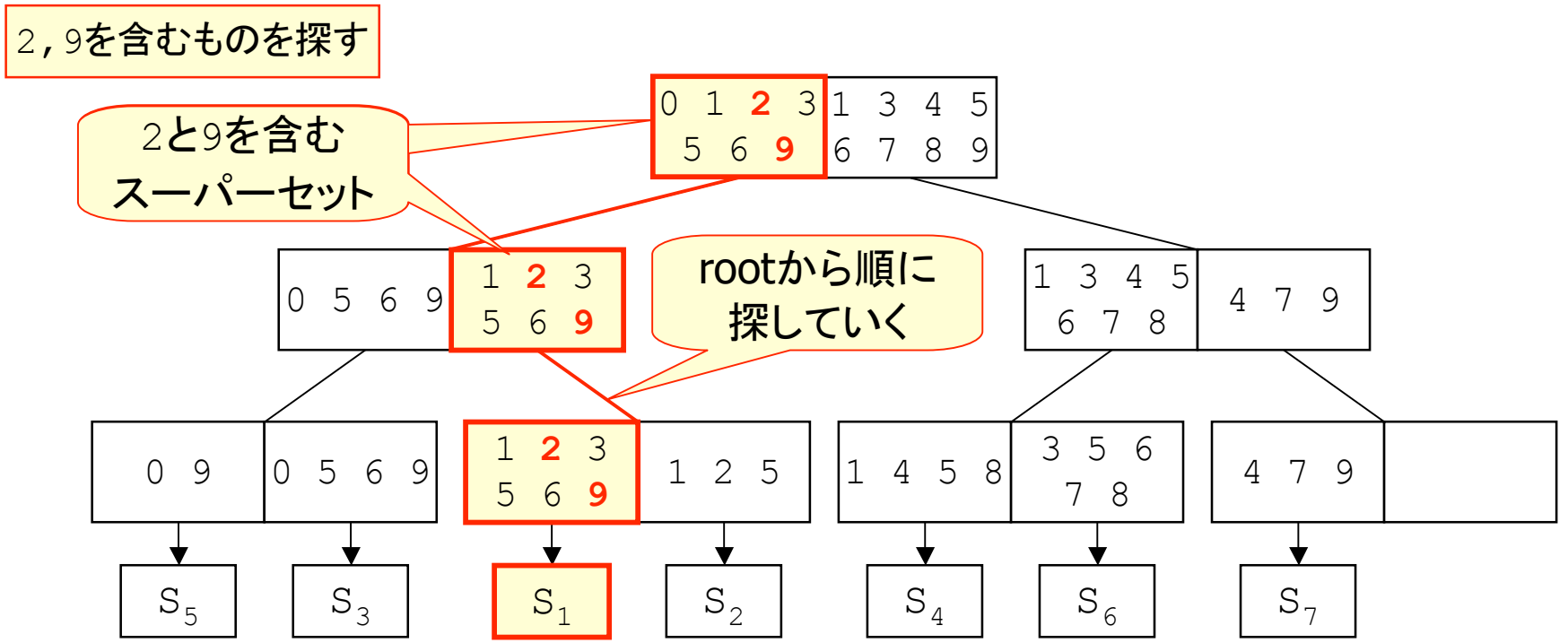


(2)N-gram方式
＝ N文字ずつ機械的に抽出

特徴：
- 漏れのない検索
- 検索されすぎる
- インデクスサイズ大

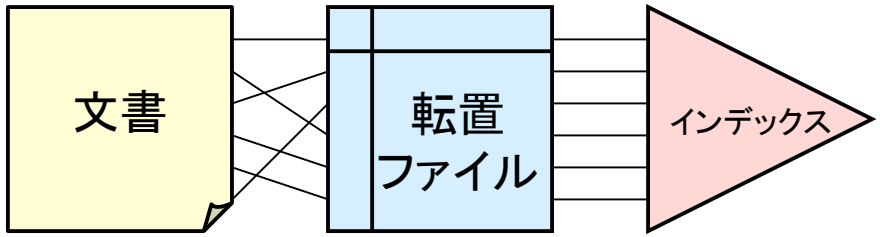
参考資料2-1: RD-Treeインデックスの概要

- Russian Doll Treeによるインデックス
  - 集合の中から指定した要素を含むものを探す
  - 親ノードは子ノードのUNIONになっている
  - R-Treeと同様のインデックス増分構築
  - 文書は固定長のビットマップシグネチャに変換して格納

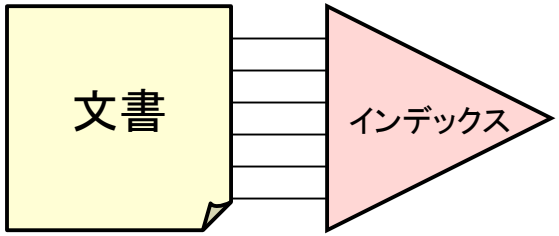


参考資料2-2: RD-Treeによるメリット

- インデックスの更新コストが低い
  - 一般的なテキスト検索では、転置ファイルにインデックスを張る
    - 転置ファイル:
      単語から文書IDを引く表
    - 文書を1つ追加すると、
      転置ファイルの多くの部分に変更が生じ、
      それにあわせてインデックスも多数更新される



  - RD-Treeでは、文書に直接インデックスを張る
    - 文書を1つ追加すると、
      その分のノードを追加するだけ
    - 但し、固定長ビットマップシグネチャへの
      変換による問題として、
      文書量が多くなると判定ミスを
      発する恐れが有る



デモ

- 8.1.0での動作デモ

公開してみて

- MLではまったく反応なし
  - あんまり全文検索ニーズってないのかしらん
- tsearch2本家MLでは多少反応あり
  - 「日本語の言語的特徴をちょっと説明してよ」と言われて激しく困った
- 問い合せはぽつぽつ
  - 90%が「UTF-8ﾏﾀﾞｰ?」

今後

- UTF-8対応
  - MeCab版、も作りかけてましたが以下略
  - kakasiもβでUTF-8対応開始
  - ただ、本家のOlegは「tsearch2はUTF-8対応してないよ」と言ってる。問い合せしてみたけど返事がない。
- tsearch2のparserとしてちゃんと実装したい
  - 単語の種類とかを考慮した検索ができるかも
    `http://www.sai.msu.su/~megera/postgres/gist/tsearch/V2/docs/HOWTO-parser-tsearch2.html`
- フルスクラッチで自作
  - DBに直接入れるのだから漏れなく検索したい
  - N-Gram方式による実装が必須