

EsTerra 

EsTerra2004

Massive & High Performance

in **MedXMLコンソーシアムプログラマーズキャンプ**



株式会社メディアフュージョン概要

● 当社概要

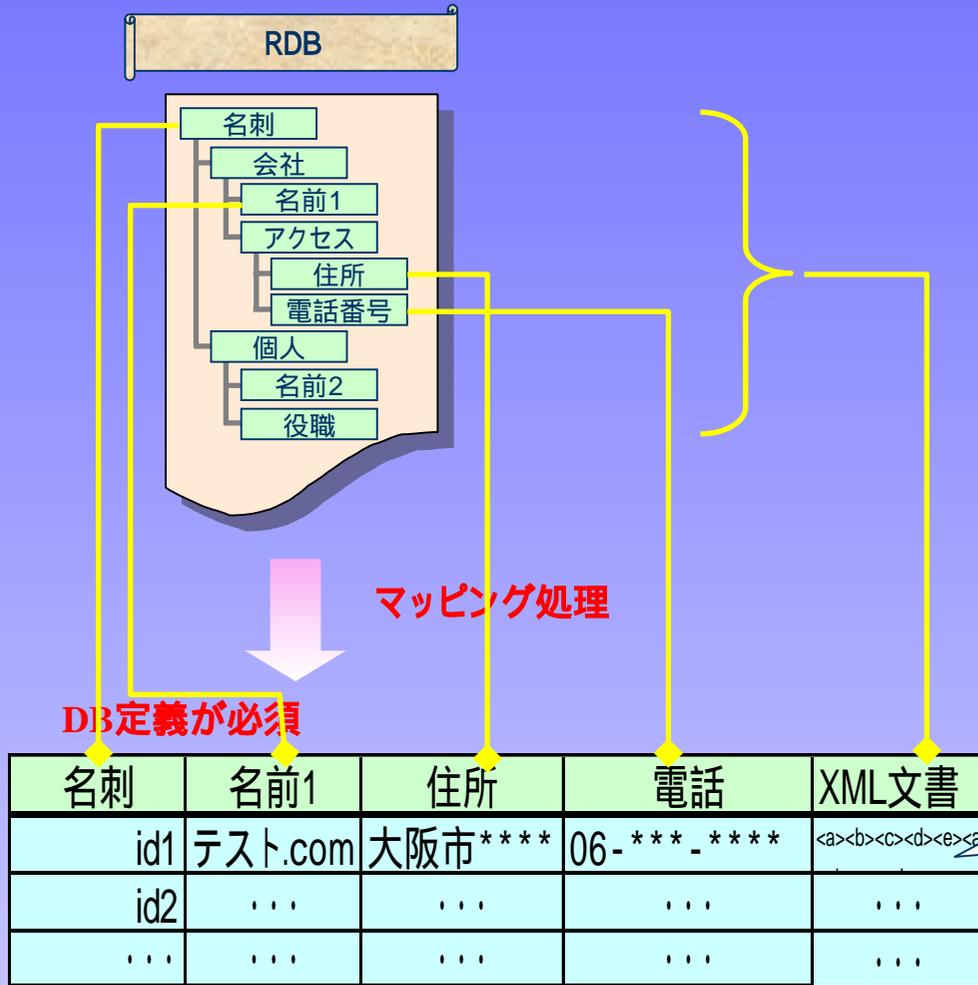
名称: 株式会社メディアフュージョン
設立: 1995年8月
資本金: 3億0930万円
本社: 大阪市北区
事業所: 東京、名古屋
 米国法人 (シリコンバレー)
社員数: 40名 (2003年5月現在)
URL: <http://www.mediafusion.co.jp>

主力製品:
ネイティブXMLストレージ「EsTerra XSS」
及び
各種Webアプリケーション開発ライブラリー・ツール

● 当社製品の医療・医薬分野での導入事例

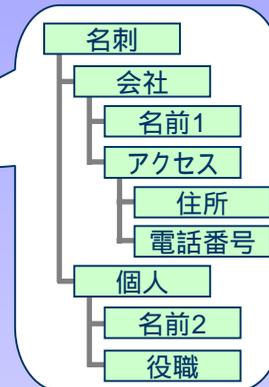
亀田総合病院様 電子カルテ退院サマリーシステム
電子カルテシステムと連携して稼働する退院サマリーを
録するシステム
電子カルテ検索システム (開発中)
電子カルテシステムからMML形式で保存し、各種の検索・
分析を行うシステム
A社電子カルテパッケージシステム (試験運用中)
開業医向けの電子カルテシステム
B社電子カルテパッケージシステム (開発中)
特定診療科専用の開業医向けの電子カルテシステム
C社医療情報データウェアハウス (開発中)
医療情報を様々な角度から分析するシステム
トレーサビリティシステム (開発中)
薬剤などのオーダリング履歴を管理するためのシステム
東京薬科大学様 医薬品副作用情報公開システム
医薬品添付文書SGML / DTDをPMLへ変換・格納し、
XPathを利用した検索を行うシステム
D社医薬品情報ワンソースマルチユースシステム
医薬品の製品情報をデータを一元化して印刷物とWebで提
供するシステム

RDBにXMLを格納する一般的な手法



課題

- XMLの構造に合わせてDB定義が必要
データ構造が変化するとDB再構築
- マッピング処理にコストがかかる
- 特定の項目のみインデックスをつけるため検索項目が限定される
- 検索にヒットしたフラグメントのみを取得できない
- ノード単位のXMLの操作は困難

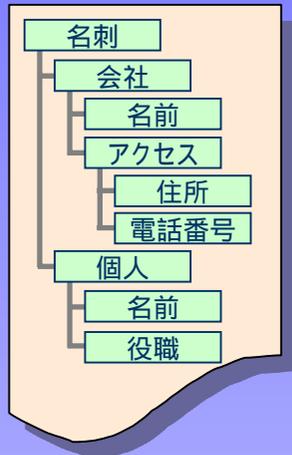


この部分は、
検索できない！
または
検索できても非常に遅い

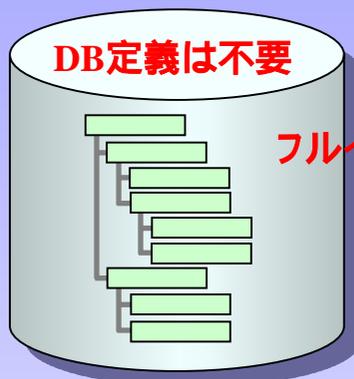
インデックスを設定した項目しか検索できない

ネイティブXMLストレージEsTerraによるXMLの格納

Well-formed XML 完全対応
ネイティブ XML DB



ツリー構造を
そのまま登録



➤ XML格納時の構造定義は不要

➡ データ構造に変更があってもDB再構築は不要

➤ 検索用インデックスの設定不要

➡ 全ノードに自動的にインデックス設定

➡ あらゆるノードに対する検索が可能

➤ Well-formed XML 完全対応

➡ DTD, XMLスキーマによる定義は全く不要

➡ RELAX NGによるバリデーションを標準装備

➤ XMLツリー構造を最大限に活かした操作

➡ ノード(エレメント)単位の操作、ロック、アクセス制限

➡ 完全なDOM操作がDB上で可能

➤ クエリ言語としてXPath, XQueryを使用可能

MMLベンチマークテスト条件

◆ 動作環境

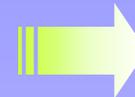
OS:Windows 2000 Professional
 CPU:Xeon 2.5GHz
 Memory:1.5GB
 HDD:1.8TB (ADTX AXRR-P250SS SCSI Disc Device)

◆ 検索式

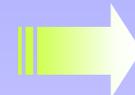
- ・次の式を4ステップに分割して実行
- ・ヒット件数は徐々に増加する結果となる

```

/root/levelone[clinical_document_header/patient/person/id[
(@EX="861047" and @RT="3.76.327.6.844170.6.306") or
(@EX="478217" and @RT="6.33.486.6.348472.6.739") or
(@EX="603414" and @RT="8.2.378.1.777992.6.574")]]
/body/section/paragraph/content/local_markup/docInfo/title[
contains(generationPurpose,"Rad") or
contains(generationPurpose,"mma") or
contains(generationPurpose,"Adm") or
contains(generationPurpose,"Inp") or
contains(generationPurpose,"Con") or
contains(generationPurpose,"Dis") or
contains(generationPurpose,"Out") or
contains(generationPurpose,"Tes") or
contains(generationPurpose,"Rec")]
  
```



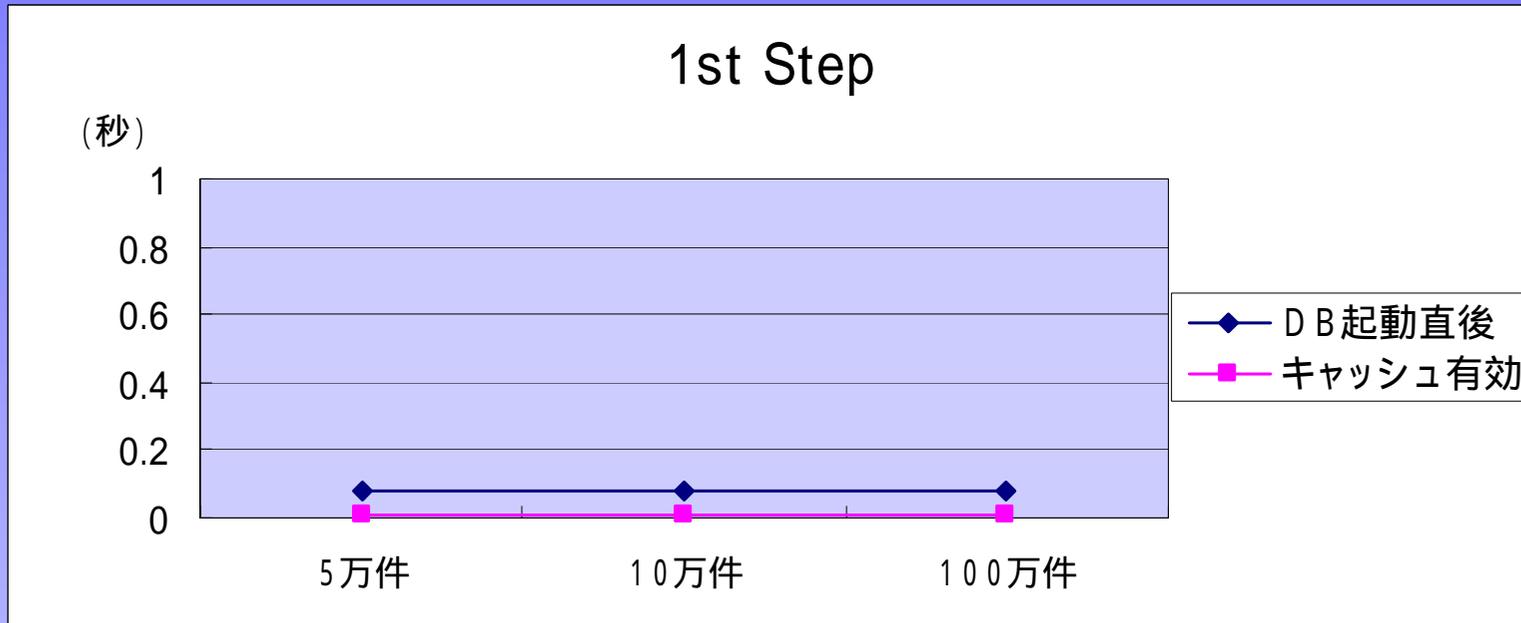
患者ID(@EX),施設ID(@RT)
 のandをとったもの3つをorする



文書詳細種別9つをorする

MMLベンチマーク結果1 (1st Step検索)

1st Step /root/levelone[clinical_document_header/patient/person/id]
 (@EX="861047" and @RT="3.76.327.6.844170.6.306") or

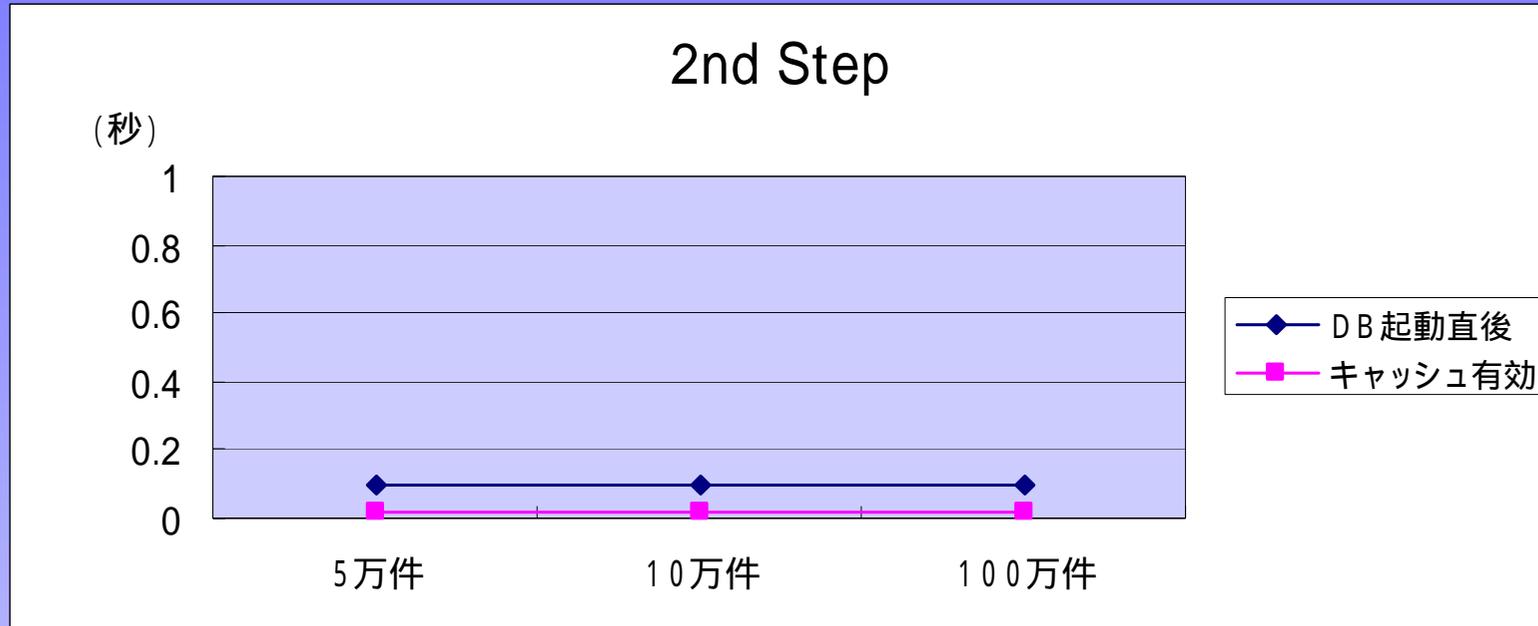


	DB 起動直後	キャッシュ有効
5 万件	0.081 秒	0.010 秒
10 万件	0.081 秒	0.010 秒
100 万件	0.080 秒	0.010 秒

データ量に依存しない高速性

MMLベンチマーク結果2 (2nd Step検索)

2nd Step (@EX="478217" and @RT="6.33.486.6.348472.6.739") or

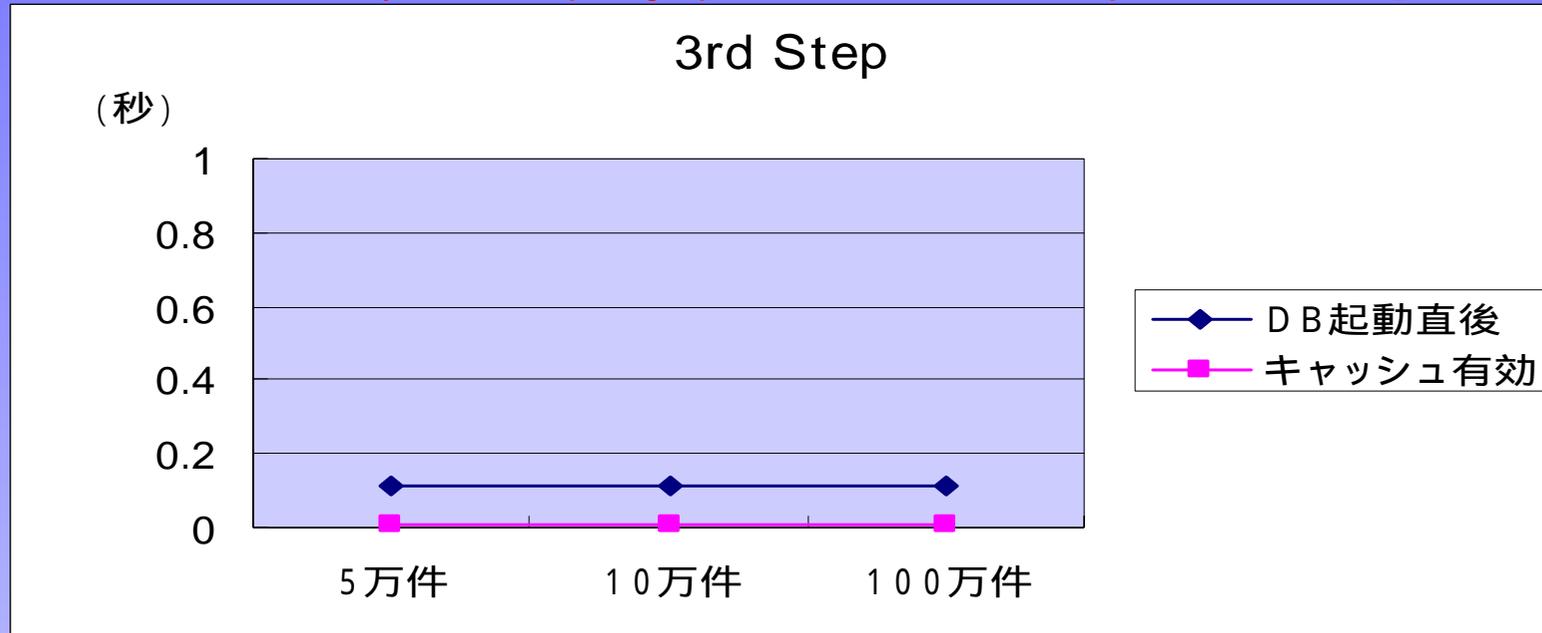


	DB起動直後	キャッシュ有効
5万件	0.094 秒	0.015 秒
10万件	0.094 秒	0.015 秒
100万件	0.094 秒	0.015 秒

データ量に依存しない高速性

MMLベンチマーク結果3 (3rd Step検索)

3rd Step (@EX="603414" and @RT="8.2.378.1.777992.6.574"]]
/body/section/paragraph/content/local_markup/docInfo/title[



	DB 起動直後	キャッシュ有効
5万件	0.112 秒	0.010 秒
10万件	0.112 秒	0.010 秒
100万件	0.110 秒	0.010 秒

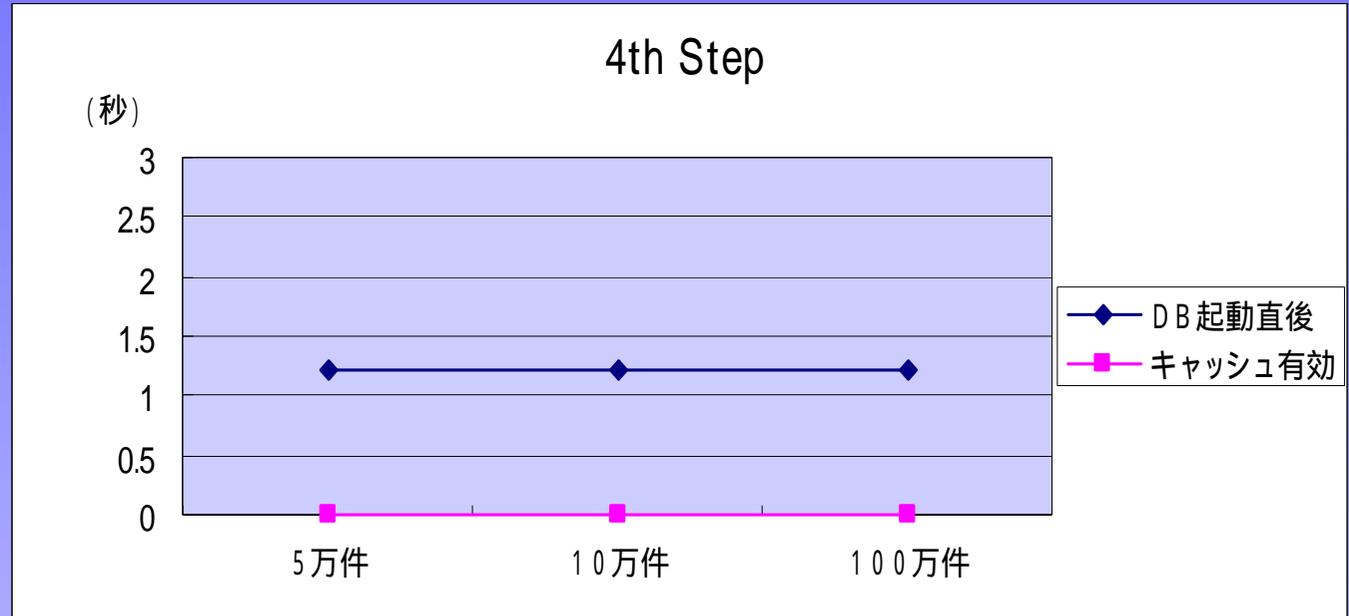
データ量に依存しない高速性

MMLベンチマーク結果4 (4th Step検索)

4th Step

```
contains(generationPurpose, "Rad")
or
contains(generationPurpose, "mma")
or
contains(generationPurpose, "Adm")
or
contains(generationPurpose, "Inp")
or
contains(generationPurpose, "Con")
or
contains(generationPurpose, "Dis")
or
contains(generationPurpose, "Out")
or
contains(generationPurpose, "Tes")
or
contains(generationPurpose, "Rec")]
```

複雑な部分一致検索
でも高速検索



	DB 起動後	キャッシュ有効
5万件	1.220 秒	0.010 秒
10万件	1.220 秒	0.010 秒
100万件	1.218 秒	0.010 秒

1000万件の電子カルテデータによる検証(参考)

某大学病院での実装を想定した

1000万件の電子カルテデータに対するパフォーマンス検証の結果

検索方法	検索式	平均検索時間
患者IDによる検索	/root/data[@PID="xxxx"]	0.003秒
患者氏名による検索	/root/data[@PID-NAME="yyyy"]	0.008秒
患者IDによる検索	/root/data[@DOCDATE>="xxxxx" and @DOCDATE<="yyyyy"][@診療科コード="aa"][@入院外来区分="bb"]/labo/[(RESINAME="cc" and (@RES-VALUE>="dd" and @RES-VALUE<="ee")) or (RESINAME="ff" and (@RES-VALUE>="gg" and @RES-VALUE<="hh")) or (RESINAME="ii" and (@RES-VALUE>="jj" and @RES-VALUE<="kk"))]/text()	0.017秒

総括

EsTerra XML Storage Server の優位点

- ◆ **Well-Formed XMLを高速処理**
 - あらゆるXML形式の医療情報を高速処理
- ◆ **データ量に依存しないパフォーマンス**
 - 膨大な医療情報を高速に検索
- ◆ **フルインデックスなので、あらゆる項目を検索可能**
 - データウェアハウスのように、様々な角度から医療情報を検索
- ◆ **データベース側の構造定義が不要**
 - 冗長性の高い医療情報を柔軟に格納、開発工数の削減・拡張性

デモ内容

- ◆ **MMLデータ検索の実際(リソースの限られたノートPCでの検証)**
 - 5万件データから検索
 - データ構造を変更させた検証結果(DB定義不要の検証)
- ◆ **Web Application Generatorの紹介**
 - Webアプリケーション自動生成(ノンプログラミング)
- ◆ **医療情報知的検索の紹介**
 - 自然文から検索用メタXMLを自動生成して、それを検索することにより所見の検索を行う。

	デモ用ノートPC動作環境
OS	Windows 2000 Professional
CPU	Pentium 750 MHz
Memory	256 MB
HDD	20GB