生命科学データベース横断検索の内部データ最適化

○大波純一[1]、杉崎太一朗[2]、友田史緒里[2]、牧口大旭[2]、川本祥子[3,4]、畠中秀樹[1,3]

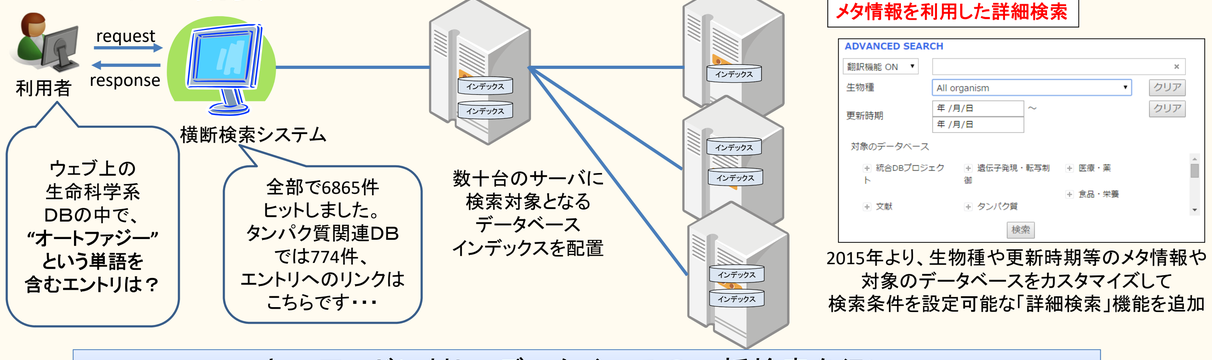
1.国立研究開発法人科学技術振興機構バイオサイエンスデータベースセンター（NBDC）、2.三井情報株式会社、
3.大学共同利用法人情報・システム研究機構ライフサイエンス統合データベースセンター（DBCLS）、4.情報・システム研究機構 国立遺伝学研究所

要旨

生命科学系データの検索による発見は、研究データの活用やデータベースでの公開が一般的になった現在でも重要なプロセスである。生命科学データベース横断検索では10年以上前からこのような検索基盤を提供し続けている。この一方で内部の検索用データ（インデックス）の構築・保持方法は、様々な体制変更やハードの設計、連携機関との協調状況、検索アルゴリズムの改善に合わせて常に変化を続けてきた。ネットワークを介した分散配置か1か所に集約する方式か、ミラーリングや冗長化、シャーディングはどのように行うか、分かち書きや利用する辞書はどのようにするか、N-gramの解析モデルはどのようにするか、などを検索パフォーマンスや検索結果のランキングモデルの需要に合わせて、調整を続けている。大量で多様な生命科学系データの検索基盤として要求されるインデキシング方法に関するこれまでの経過をまとめ、最適な構成について改めて検討を行った。

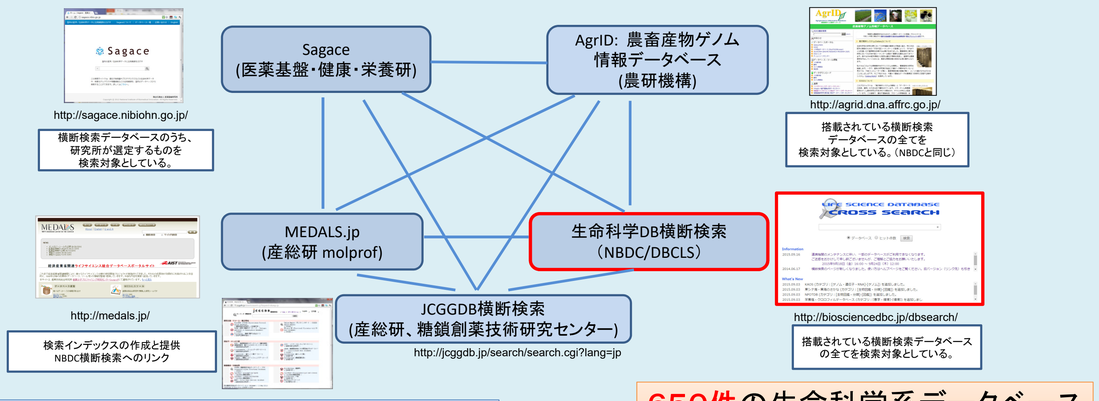
生命科学データベース横断検索とは

システム概要



キーワードに対し、データベースの一括検索を行い、
ヒット件数と、検索結果（対象へのリンク、ヒット箇所）を返す**Web検索システム**

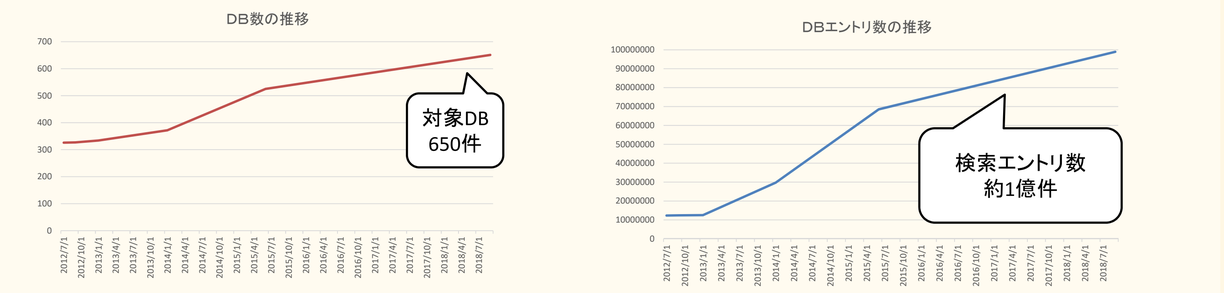
組織間連携



4省庁の別組織にサーバを分散配置し、
相互に参照しながら検索可能。

650件の生命科学系データベース
約1億件のエントリを対象に
検索可能　※2019年10月時点

対象データの増加



検索対象の増加、ネットワーク的に離れた組織との接続、クラスタマシン構成

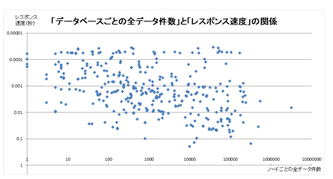
検索のために最適な内部データの配置と構築方法は何か？

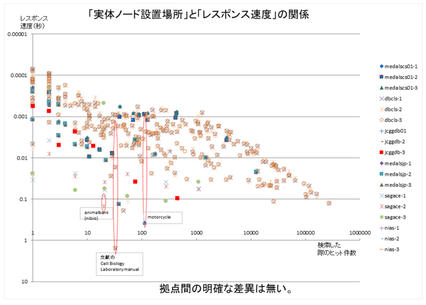
Hyper estraier（HE）の場合

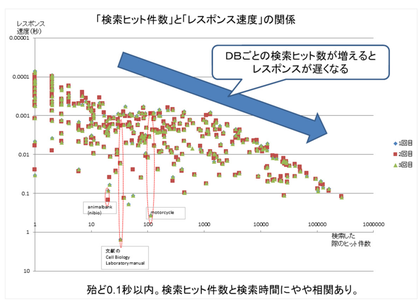
2008年に生命科学データベース横断検索が構築された際、選定された検索エンジンアプリケーションは「Hyper estraier」。
このアプリケーションは「P2Pでネットワークを介した分散配置可能」、「日本人の開発者が作ったため日本語のインデキシングが特別なチューニング無しに可能」、「スケーリングで大量データを横断検索可能」という特徴があり、初期要件を満たすものと考えられていた。
デフォルトの設定で2.2-gramのインデックスを作成していたが、1文字検索の結果が抜けるため1.2-gramとなるように修正して、転置インデックスを構築した。

発生した課題：　検索パフォーマンスの低下

当初は高速な検索を実現していたが、検索対象が増えるごとに検索速度が低下し、ヒット数が多いと1回の検索レスポンスに10秒以上要するようになってしまった。下の4図は「e」で検索した場合の検索時間を示す。

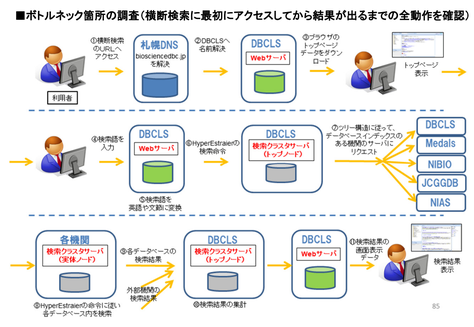


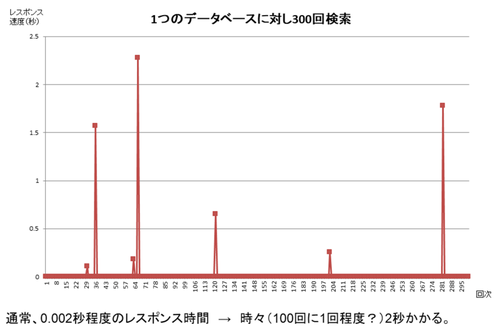




分析と対策

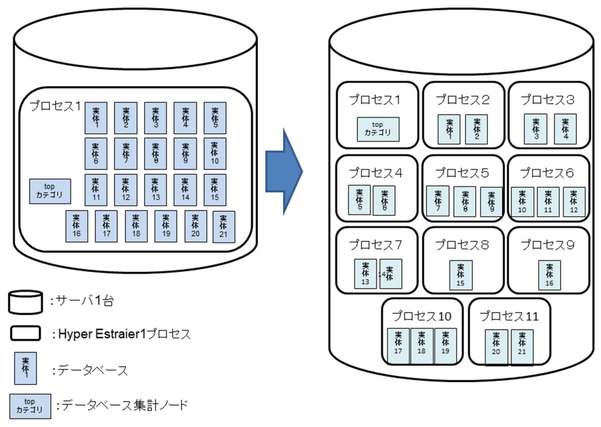
問題箇所を確認するため、ボトルネックの洗い出しの通信テストを行った（左下図）。この結果、DB1つで構成されているインデックスごとに1/100程度の割合で遅延が発生していること（右下図）、検索結果のマージ時に遅い検索結果を待っているために、当時400DBの横断検索では必ず遅くなってしまうことが分かった。メモリキャッシュのリフレッシュを定期的に行うことで状況が改善した。





通常、0.002秒程度のレスポンス時間　→　時々（100回に1回程度？）2秒かかる。

さらに内部データ構成やプロセス運用の検討を実施した結果、①数10万件以上のエントリを含むDBインデックスは数万件ごとに分割して、複数のDBインデックスにした方が速いこと、②1つのサーバ内に複数のHEプロセスを別ポートで立ち上げた方が、メモリ利用量を限界まで使うことができ、効率的に検索できることを確認した。



最終的に左図のように、DBの分割とプロセスの複数起動を実現し、検索速度が改善した。

このプロセス分割では、HEの通信ポートがデフォルトの1978だけであったものが、新たに10種類必要となり、ネットワークを介した連携機関には、ファイアウォール疎通設定などの運用負荷がかかることとなった。

この後、メモリリークが発生してHEのプロセスが落ちるようになり、落ちたプロセスを検知して自動的に再起動するツール（monit）を導入した。

Elasticsearch（ES）の場合

ハードの動作に関与する内部データの設定

HEが開発終了となり、検索速度の限界が見え始めたため、2015年に検索エンジンアプリケーションを「Elasticsearch」に変更した。初期段階ではまずインデックスをNBDC内のサーバー1か所にまとめ、徐々に分散化しスケールアップしていく手順を検討して構築したが、予想以上のパフォーマンス低下が見られ、調査と対応を実施した。

①ディスクサイズ・物理メモリサイズ・ヒープメモリサイズの最適化

アプリケーション移行直後の横断検索のハード環境は右表のような構成で稼動しており、ディスクもメモリも最大限活用できるように配置していた。
ESを動かしているJavaの構造から、記憶装置の容量の比率は「ディスクサイズ：物理メモリ：ヒープメモリ = 48：2：1」が最も効率良く動作することが分かり、データの再配置やメモリ使用量の調整を実施した。

CPU	Intel(R) Xeon(R) CPU E5-2650 v2 @ 2.60GHz, 4Core
メモリ	29GB
HDD	292GB
OS	CentOS 6.2
台数	20台

②shardの設定

ESでは、冗長性確保と高速な検索プロセスのために、「shard」と呼ばれる単位で検索データのコピーを作成し、マスタデータは「primary shard」、セカンダリ以降のデータは「replica shard」として複製される。ESの通常の設定ではreplica shardが一つできることになるが、この時にディスクを2倍消費することになる。さらに複製されたshardはクラスタ構成となった複数サーバ内を、RAIDのストライピングデータのように分割されて、全体のディスク使用量がほぼ均一になるよう動的に移動する。これらの仕様が検索エンジン移行後に判明し、数百GBがサーバ間をダイナミックに移動し通信帯域が圧迫されてしまった。最終的にreplicaの数を少なくし、indexの単位容量を少なくする（後述）ことで、shardの移動単位も小さくすることができ状況改善した。

③indexの設定

ESで直接的な検索対象となるindexは、検証時点で609.8GBあり、replica shard分も含めると1.2TBだった（右表）。indexはHEの時代から1つにまとめられるものであったためindex=1としていた。しかしこれだと②のshardの動作において、できるだけindexを大きな塊で一つにまとめたままにしておく機構が働き、サーバ間を50～100GBのデータの塊が動くことになり非効率かつ不安定であった。

そこでindexを15個に分割し、さらに分割したindexの大きさを均一になるよう調整することで、サーバ間を動くデータを数GBまで抑え、スムーズなデータ移動と、サーバのパフォーマンス向上に繋がった。

health	status	index	pri	rep	docs.count	docs.deleted	store.size	pri.store.size
red	open	nbdc	14	1	69680999	6155760	1.2tb	609.8gb

検索品質に関係する内部データの設定

ESでは検索indexを構築する時に、「analyzer」というインデキシングアルゴリズムで細かい文字単位（トークン）に分割する（tokenizerとも記される）。HEでは最終的に2,2-gramの分割基準で行われていたが、ESではより細かい指定が可能となった。

①Tokenizerの設定

HEではindex内の分割基準は2,2-gramに加えて、自動で検索データをHE内部の辞書（IPA辞書）を使い単語単位に分割（分かち書き）し、大文字小文字検索の統一にも対応していた。ESでは、bi-gramでの分割インデックス作成を実施していたが、「単語の分かち書き」と「大文字小文字検索の統一化」を意識的に設定する必要があった。このため後日「ESで提供されている日本語のkuromoji辞書を使った分かち書き」と、アルファベットのインデキシングの際の小文字への統一（lowercase）を実施した。

②ツールボックスのピックアップの設定

横断検索では全文の串刺し検索だけでなく、「検索前のサジェストボックスの検索」、「ヒット件数を表示するための検索」「ツールボックスの遺伝子名検索」、「ツールボックスの対訳辞書（京都大学のライフサイエンス辞書プロジェクトご提供）の検索」、「ツールボックスの関連語句の検索」、「ツールボックスの同義語（シソーラス）の検索」を同時に行っている。これらの中で特にツールボックスで使用する辞書のインデックスを、メインのインデックスから分離して検索結果表示の高速化を図った。

まとめ

- Hyper estraierからElasticsearchに移り変わると共に、横断検索ではパフォーマンス向上のため、内部データの最適化を継続してきた。
- 内部データの最適化に向けて、引き続き改善や強化を継続していく。

トーゴーの日シンポジウム2019 ～バイオデータベース：つないで使う～
会期：2019年10月5日(土)　於：日本科学未来館