# Apache Solr で 1 億文書の
# ファイルサーバ検索エンジンを作ってみた

株式会社 鉄飛テクノロジー
http://www.teppi.com/

# 目　次


## 作ってみた

実証結果　　---　１ヶ月でインデックス構築完了

実証データ　---　日本語 *Wikipedia* の全件データを 100 倍に増幅

実証環境　　---　数年経過した普通の１U サーバ

運用課題　　---　インデックスの差分更新に４日必要

## Apache Solr は「使える」。ただし・・・

分散検索システムによるスケールアウトの目的

目的合理性のない分散は無駄

## ファイルサーバ検索における Apache Solr の活かし方

インデックスを「分割して統合する」のがポイント

ファイル検索の索引は、ファイルサーバ別・共有フォルダ別に編成するのが現実的

## ファイルサーバ検索だから「手抜き」できること

SolrCloud 冗長構成はいらない？

## シンプルな Apache Solr なら、管理も簡単です

簡易 SolrCloud だから簡単にできる

**パッケージとして販売されているファイルサーバ検索システムの多くは、「1 千万文書の大規模検索が可能」など、その高性能をアピールしていますが、実際には何千万文書ぐらいまでの全文検索に利用できるのでしょうか。多くの製品が、ノンカスタマイズでパッケージとして対応できる文書数は、１千万文書の大台あたりであるように思われます。**
**弊社のファイルサーバ全文検索システム「FileBlog」も、3000 万文書クラスの検索まではノンカスタマイズで対応していますが、これを実証するために、実際に一桁上の「１億文書」の全文検索インデックスを構築してみることにしました。**

## 作ってみた

### 実証結果
### ～　１ヶ月でインデックス構築完了　～

・１億文書のインデックスの構築に成功しました
・インデックス構築の所要時間は約１ヶ月でした
・全文検索の実行に成功しました
・100 万文書の *Wikipedia* データの複製 100 個で 1 億文書が構成されています
・100 万文書の *Wikipedia* データフォルダ 1 つを対象とする検索の場合、初回は 20 秒ほどかかりますが、2 回目以降は 2 秒未満で高速に可能です。初回の検索時にインデックスデータがハードディスクからメモリにキャッシュされることで次回以降の検索が高速化されていると考えられます。
・全 1 億文書対象の検索時には数分程度かかります。これを高速化するには、ヒープメモリをさらに拡張する必要があるでしょう。

## 実証データ
## ～　日本語 Wikipedia 全件データを 100 倍に増幅　～

1 億文書の Windows ファイルサーバを用意して、これを対象に全文検索インデックスを構築します。ただし、実際に 1 億文書を収集することは容易でありません。そこで今回は、日本語 Wikipedia の全エントリのデータ（約 100 万件）をダウンロードしてテキストファイルとしてファイルサーバ上のフォルダに展開し、このフォルダを 100 倍にコピーして、1 億文書（ = 100 万文書 X 100 ）のファイルサーバを構築しました。（実際には容量節約のため、コピーのかわりにシンボリックリンクを用いました。）

## 実証環境
## ～　数年経過した普通の 1U サーバ　～

**●実証環境は、下記構成のホスト上に仮想マシンを作成しました。**

- VMWare ホスト　：　FUJITSU PRIMERGY RX200 S6
  - Intel Xeon X5650 CPU(6core 2.67GHz)×2 … 24vCPU
  - 96GB メインメモリ
  - 900GB SATA SSD ×2
  - 大容量 iSCSI ドライブ(1000Gbps 接続)

●仮想マシンは 2 台

・検索エンジン(Apache Solr)を主に稼働させる検索用ノード
  - ４VCPU メモリ 8GB　仮想ハードディスク約 100GB

・FileBlog システムを稼働させるノード
  - 2VCPU　メモリ 32GB 仮想ハードディスク（システム 40GB/データ 1TB）
  - Apache Solr の Java ヒープメモリ 22GB

## 運用課題
## ～　インデックスの差分更新に 4 日必要　～

●実証環境では 1 億文書のインデックスの差分構築の実行には 3 日～4 日かかりました。

●日常運用継続のためには、差分インデックス更新を週末 48 時間以内程度に高速化する必要がありますが、適切なスケジューリングやハードウェア選定によって十分実現可能な範囲でしょう。

# ファイルサーバ検索に Apache Solr は「使える」。ただし・・・

Apache Solr は、オープンソースの全文検索エンジンとして、現時点でデファクトスタンダートとなっている製品です。鉄飛テクノロジーのファイルサーバ検索システム「FileBlog」も Apache Solr を組み込んで動作しています。

## ■分散検索システムによるスケールアウトの目的

Apache Solr が標準で用意するスケールアウト機能が SolrCloud です。

コンピュータシステムで、大量のデータを対象とする処理の性能を引き上がるための手段として、複数のマシンを並べて処理を行うことを「スケールアウト」と言います。(なお、単一マシンにおいて、CPU のグレードアップやメモリの増設/ディスク増設によって性能を引き上げる「スケールアップ」と対比される言葉です。)

検索システムにおいてのスケールアウトの目的は、下記の２点となります。

① 複数マシンを束ねて大量リソースを利用可能にし・・・

→ 対象ドキュメントの数量が大きい大規模検索を可能にする、巨大な検索インデックスを構築・維持可能にするため。

＊ 実際に 1 台のサーバに配置できないような巨大な全文検索インデックスも、shard と呼ばれる単位に分割して、複数台のサーバに分散配置することを可能にしています)

→ 多数のユーザからの、同時多数検索クエリ要求に速やかに応答するため。

＊ 1 台のサーバでは応答性能に限界がありますが、インデックスのデータを分散配置させた複数サーバによって、同時に多数のクエリ要求に応答できるようになります)

② 検索インデックスのデータを複製して分散配置する、「冗長化」によって・・・

→ ハードウェア障害による検索システムの停止リスクを小さくするため。

＊ インデックスデータのレプリカを持つノードを複数サーバに分散配置することで、いずれかのノードが障害で停止してもシステム全体は無停止で稼働しつづけるように構成することが可能です)

## ■目的合理性のない分散は無駄

「SolrCloud を使って、複数マシンの処理を分散する」と言うと、何か 1 台のマシンでやるよりもすごいことのように聞こえるかもしれません。しかし、 1 台で可能なら、 1 台で実現するほうが、間違いなく簡単で安上がりです。
例えば、下記のような「無意味なスケールアウト」はやめましょう。

***「スケールアップできるなら、２台に分割せず増強すべき」***

近年の IT 技術の進歩は著しいものです。普通の PC サーバに詰めるメインメモリの量も、ずいぶんと大きくなりました。たとえば、2018 年 1 月現在、普通の 1U ラックサーバでもメモリスロットが 16 本はあるでしょう。32GB×16=512GB あるいは 64GB×16=1TB のメモリを搭載できる計算です。64GB メモリのマシンを 8 台調達するより、512GB メモリのマシン 1 台で済ませたほうが当然、安上がりです。

***「仮想化環境でスケールアウトするのは、本末転倒」***

もともと処理負荷の比較的小さかった多数の物理サーバをそれぞれ仮想マシンとして、少数の高性能な（スケールアップされた）サーバ上に集約する、というのが「仮想化」環境構築の本来の動機です。1 台で処理できない高負荷処理を多数のハードウェアで実現する「スケールアウト」とは完全に逆のアプローチです。

既存の仮想サーバインフラ上に、あえて複数の仮想マシンを構築して処理を分散させるよりも、できるだけ高性能の仮想マシン 1 台を構築して処理させたほうが高性能になる可能性が高いでしょう。利用可能な総リソースが同じであるので、あえて分散させる理由はあまり見当たりません。

***「データアクセス（ファイルサーバ）が遅ければ、がんっぱっても性能は出ません」***

ファイルサーバ検索エンジンの性能は、ファイルサーバの性能に依存します。ファイルサーバ上のデータに対するアクセス性能が低ければ、検索エンジンのインデックス構築の所要時間はどうしても長くなってしまいます。ファイルサーバのディスク性能にかぎらず、ファイルサーバと検索サーバ間のネットワーク性能も、ファイルサーバ全文検索システムの実用性能に大きく影響します。

たとえば、ファイルサーバと検索サーバ間のネットワークが 1Gbps の環境で、どうしても検索インデックス構築が遅いという場合があるかもしれませんが、10GbE ネットワークが普及すれば、このボトルネックが一気に解消されることでしょう。

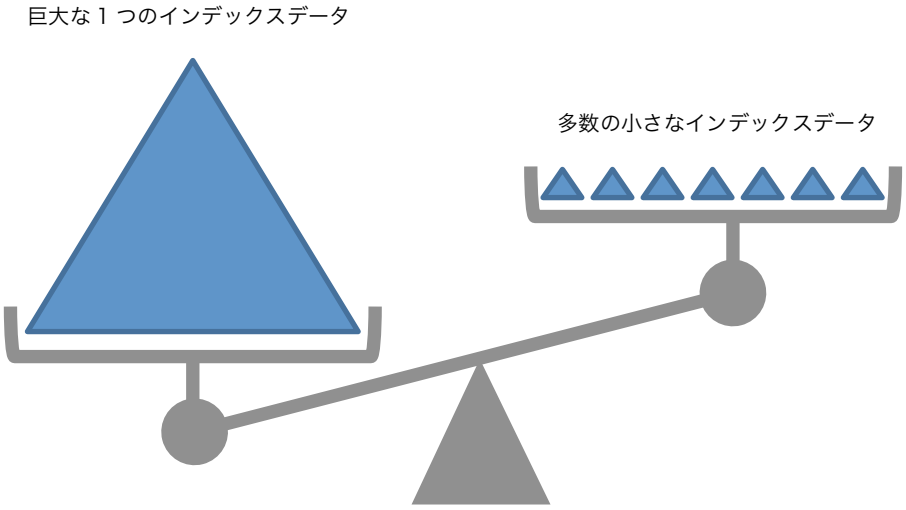
また、全文検索エンジンは起動時にインデックスデータの読取りを行うため、インデックスのデータ量に比例した（再）起動時間が必要です。数千万文書のインデックスをハードディスクから読み取るだけで、起動処理に数十分を要することもありえますが、インデックスデータが SSD 上に格納されていれば、その数十分を数分に短縮できたりするのです。

SolrCloud による分散構成は、性能問題に対する特効薬ではありません。時間が許すなら、安くて遅いディスク・ネットワークでも構いませんが、予算が許すなら、速いディスク／速いネットワークを選択して後悔することは無いでしょう。

# ファイルサーバ検索における SolrCloud の活かし方

## ■インデックスを「分割して統治する」のがポイント

検索エンジンは対象文書のテキストを読み込んで、インデックス（索引情報）を構築しますが、このインデックスの作り方次第で、検索性能には大きな差が生まれます。たとえば同じ 1 億文書でも、「1 億文書の巨大な索引×1 個」よりも「100 万文書の小さな索引 × 100 個」の方が断然扱いやすいことは、見逃せない事実です。




分割して統治すれば（処理が）軽くなります

## ■ファイル検索の索引は、ファイルサーバ別・共有フォルダ別に編成するのが現実的

ファイルサーバ検索をご検討中のお客様の場合、一つのファイルサーバの一つの共有フォルダだけでファイル数が数千万文書に及ぶことはまれでしょう。実際には、数百万文書・1TB～2TB 程度の共有フォルダが数十個あって、合計で数千万文書クラスの容量になっていることのほうが一般的です。こういう場合、数千万文書全体で一つのインデックスを構築するよりも、個別の共有フォルダ・個別のファイルサーバごとに小さなインデックスを構築するほうが都合が良いのです。その理由は以下の通りです。

- 実際の検索は、全共有フォルダからではなく、特定のファイルサーバ内でのみ行われることがほとんどではないでしょうか？特定の共有フォルダを選択してから検索するというユーザインターフェイスがあれば、より少ないハードウェアリソースで、より高速に検索することが可能です。検索対象を全共有フォルダとする「全ファイルサーバ横断検索」も実現できるに越したことはありませんが、そのためにハードウェア予算や運用費用が倍増する程の価値があるのか、考え直してみてはいかがでしょうか？

- 多くの企業において、ファイルサーバの構成は決して固定的ではありません。拠点の新設、部門の統合、部門の分割など、組織構造の変化は必ず発生するのが現実です。ファイルサーバもその影響を受けずにはすみません。共有フォルダの分割・統合が発生したとき、共有フォルダ・ファイルサーバごとにインデックスが独立していれば、その小さな単位でインデックスを初期化して作り直したり、破棄したりすることで、検索エンジン全体のインデックスには大きく影響を与えることなしに、インデックスの保守が可能になります。

## ■1 台のマシンでも SolrCloud にする意味がある

一般的には SolrCloud 環境といえば何台ものサーバを並べて使うものと理解されていますが、検索インデックスを「分割して統治する」という目的のためには、たとえ 1 台のマシン上であっても SolrCloud 環境を構築することが意味を持ちます。FileBlog では数千万文書のファイルサーバ検索を 1 台のマシンで実現可能ですが、数千万文書にのぼる文書をインデックスする場合、1 台のマシンでも SolrCloud 環境を構築して運用することを推奨します。

# ファイルサーバ検索だから「手抜き」できること

## ■SolrCloud 冗長構成はいらない？

SolrCloud は、24 時間 365 日連続稼働を宿命付けられたミッションクリティカルなシステムでの利用を可能にするために進化してきた分散検索システムです。検索インデックスを複製して多数のノードに配置する（冗長構成を取る）ことで、少数のノードにハードウェア障害がおきても検索システム全体としてはデータを失わず、機能を停止すること無く稼働を続けることが可能です。

冗長構成にはもちろんメリットがありますが、ハードウェアを余計に必要とするため、環境構築の費用は間違いなく高くつきます。FileBlog の目的である「ファイルサーバ全文検索」は、24 時間 365 日、１分間の停止も許されないほどの要件では必ずしもありません。ファイルサーバ検索エンジンにそこまでの可用性は、やりすぎではないか？と私たちは考えました。

***「レプリカを１つしか持たない構成が可能です」***

レプリカを持てばデータの安全が保てます。SolrCloud の教科書的な資料を見ると、いずれもレプリカを複数ノード上に持ってインデックスデータを冗長化している例が目立ちます。

しかし、ファイルサーバ全文検索システムにおいては、ファイルサーバ上の元ファイルさえ無事であれば、検索システムが持つインデックスデータに障害が発生しても、元ファイルを参照してインデックスデータを再構築することが可能です。ただただ時間がかかるかもしれませんが、時間だけの問題です。

そこで、FileBlog の SolrCloud 管理機能では、デフォルトでレプリカを持たないで検索インデックスを構築するようにしています。

RAID には JBOD/RAID0/RAID1/RAID5 などいろいろがありますが、冗長性の無い RAID0 や JBOD も用途に応じて利用されているように、SolrCloud でもレプリカ無し構成を使う意味はあるのです。

## ■Zookeeper を 1 台の簡易構成に省略できます

SolrCloud を構成する場合、複数ノードが協調して動作するためのコーディネイトを行うためにミドルウェアとして zookeeper が使用されます。無停止が求められるミッションクリティカル環境では、3 台の独立のハードウェアに zookeeper をインストールする必要があります。そうしておけば、3 台中 1 台が故障しても大丈夫です。

しかし、ファイルサーバ検索を実現するために、FileBlog サーバと合わせて 4 台ものマシンが必要というのでは、現実的ではありません。

実際には FileBlog が稼働するサーバ 1 台に zookeeper を同居させ、1 台構成の zookeeper で SolrCloud を立ち上げることが可能です。開発環境・テスト環境など、高可用性が求められない環境に限定される構成ですが、この構成で本番運用してはいけないなんていうルールはありません。

zookeeper を 1 台だけで稼働させた場合、その 1 台がクラッシュすれば SolrCloud としては機能不全状態となりますが、個別の全文検索エンジンは動き続け、インデックスは無傷で残ります。広範囲にわたる横断的検索は不可能となりますが、復旧は時間の問題です。

高可用性を諦めれば、***3 台少ないマシンで SolrCloud を構築できる***のです。

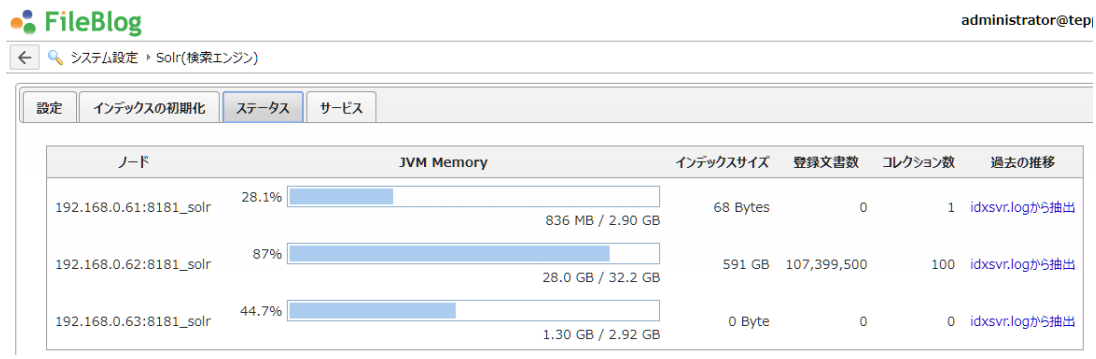
# シンプルな SolrCloud なら、管理も簡単です

一般的に SolrCloud 環境の構築は、単体の Solr サーバ構築と比べて格段に難しいものです。

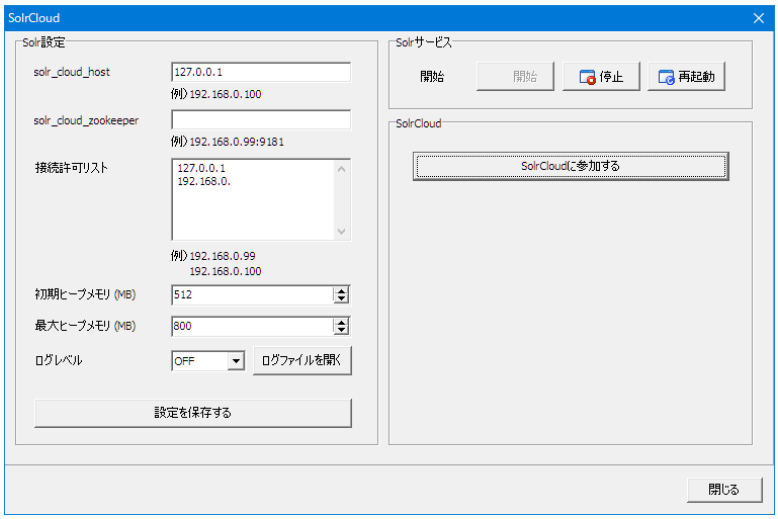
業務で SolrCloud を立ち上げようと思うと、まずは SolrCloud について書かれた解説記事や勉強会の資料を読み込んでおおよそを理解した上で、Solr の英語ドキュメントをリファレンスとして参照しながら、実際に Solr 環境を作ってみて、その挙動を一通りテストして把握する、という手順を踏み、何週間も費やさなければソリューションの評価もできないのが普通でした。パッケージシステムに組み込むにしても、有償の導入支援サービスでベンダーのエンジニアが Solr 環境を手動で構築するために、予算規模が大きくなりがちでした。

私達は、ファイルサーバ全文検索システム「FileBlog」に Apache Solr を組み込み、セットアッププログラムを実行するだけで必要な環境が全自動でインストールされる仕組みを、10 年ほど前から維持しています。大規模分散検索システムについても、Apache Solr や SolrCloud に関する一切の事前知識や試行錯誤なしに、お客様が FileBlog のインストーラとサポート文書だけで構築できるようにしたいと考え、SolrCloud 環境の構築・運用に必要なツール類をパッケージに組み込みました。

FileBlog では、たとえば以下のような管理画面によって SolrCloud の管理が可能ですので、Apache Solr の英語リファレンスを調べたり、技術勉強会に参加したりしなくても、日常的なシステム管理が簡単に画面からできるようになっています。




ノード一覧　（たとえば、複数マシンで動く Solr インスタンスを一覧できます）




ノード管理　（たとえば、新規 Solr ノードを初期化して追加できます）


コレクション管理　（検索対象の共有フォルダと 1 対 1 で対応するインデックスを管理します）

## ■簡易 SolrCloud だから簡単にできる

FileBlog が SolrCloud の環境構築を簡単に実現しているのは、24 時間 365 日データを安全に無停止で運用するというような、「高い要求レベル」をある程度捨てて、簡易構成を基準に設定画面を設計しているためです。もちろん、お客様がハードウェアにより多く投資し、より多くの運用コストをかけてもかまわない場合には、高い可用性や高い応答性能を可能にする構成も使えるようにしていますが、私たちは「簡単な構成を簡単に構築・保守できる」ことにより大きな重点を置いております。

自力ではファイルサーバ検索エンジンを構築しようと思ったが、SolrCloud は難しくて構築できなかった、というお客様でも、FileBlog で簡単に SolrCloud を利用した大規模検索エンジンが実現できるようになっていますので、ぜひ私達にご相談の上、お試しください。

２０１８年３月発行

株式会社 鉄飛テクノロジー　　http://www.teppi.com
〒153-0064 東京都目黒区下目黒 1-4-6 ロイヤル目黒 1F
TEL：03-3491-1812　　E-Mail：sales@teppi.com