高速なログ検索エンジン Hayabusa について

あべひろし (@hirolovesbeer)

2017/12/20

1　背景と目的

ネットワークのトラブルシューティングやセキュリティインシデントに対応するため，ネットワーク管理者はトラブルの原因を特定するためにサーバやネットワーク，セキュリティ機器から出力されるログを蓄積し，検索をすることがある．大規模なネットワークでは，出力されるログの量も多く蓄積・検索システムの規模も巨大化する．大量に出力される機器のログを高速に蓄積し，高速に検索するシステムとして Hayabusa[1] を実装した．

2　Hayabusa のアーキテクチャ

Hayabusa はオープンソースソフトウェアとして実装され，GitHub 上で公開されている（https://github.com/hirolovesbeer/hayabusaa）．図 1 に Hayabusa のアーキテクチャを示す．

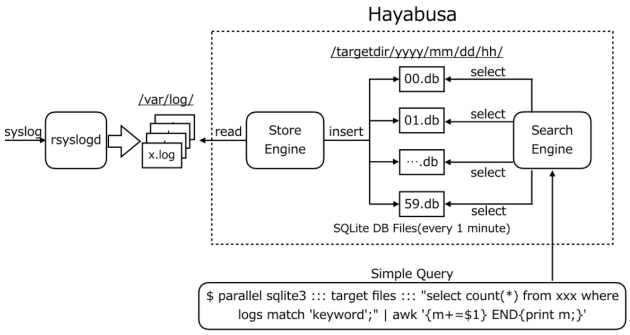

図 1: Hayabusa のアーキテクチャ

Hayabusa はスタンドアロンサーバで動作し，CPU のマルチコアを有効に使い高速な並列検索処理を実現する．Hayabusa は大きく StoreEngine と SearchEngine の 2 つに分けられる．StoreEngine は cron により 1 分毎に起動され，ターゲットとなるログファイルを開きログメッセージを SQLite3 ファイルへと変換する．ログが保存される SQLite3 ファイルは FTS(Full Text Search) と呼ばれる全文検索に特化したテーブルとして作成され高速なログ検索を実現する．

SearchEngine は，並列検索性能を向上させるために分単位に細分化された FTS フォーマットで定義された SQLite3 ファイルへアクセスを行う．各 SQLite3 ファイルへは GNU Parallel を用いて並列に SQL 検索クエリが実行され，結果は UNIX パイプラインを経由して awk コマンドや count コマンドを用いて集計される．

3　性能概要

Hayabusa はスタンドアロン環境で動作するが，小規模な Apache Spark のクラスタよりも全文検索性能が高い．性能評価実験では，Hayabusa は 3 台の Apache Spark クラスタより 27 倍早い検索性能を示した．

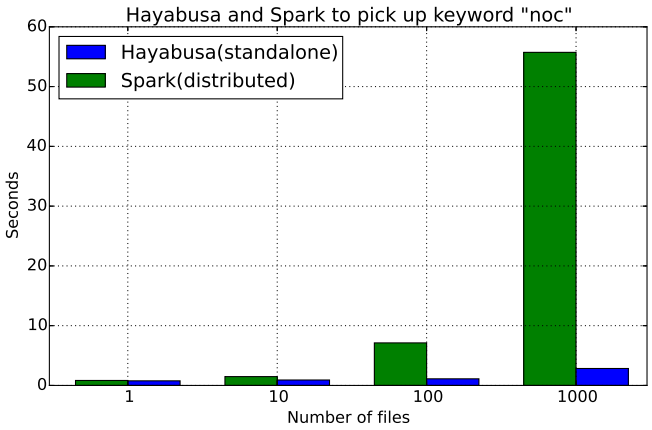

図 2: Hayabusa と Spark の性能比較

4　まとめ

スタンドアロン環境にはハードウェア限界が存在し，規模が拡大した他の分散処理クラスタにいつかは性能が抜かれてしまう可能性が高い．そこで，Hayabusa の限界であるスタンドアロン環境という制約を取り払い，複数ホストで Hayabusa の分散処理環境を構築し，検索性能がスケールアウトするアーキテクチャの実現をした [2]．計測した結果，1 台の処理ホストでは約 468 秒かかった検索処理が最大約 6 秒まで短縮した．144 億レコードの syslog データを 6 秒でフルスキャンし，全文検索可能な分散 Hayabusa 環境はログ検索エンジンとして高い性能を発揮する．

参考文献

[1] H. Abe, K. Shima, Y. Sekiya, D. Miyamoto, T. Ishihara, and K. Okada. Hayabusa: Simple and fast full-text search engine for massive system log data., CFI ' 17, pages 2:12:7, Fukuoka, JAPAN, 2017. ACM.

[2] 阿部博 and 篠田陽一, スケールアウト可能なログ検索エンジンの実現と評価, インターネットと運用技術シンポジウム論文集 2017 論文集, volume 2017, pages 73-80, nov 2017.