

# Sawmill ニュースレター

平成 20 年 8 月 19 日  
( 訳 ) ジュピターテクノロジー株式会社  
お問合せは Tel 042.358-1251,  
Email tech-support1@jtc-i.co.jp

## ニュース

Flowerfire 社は最新 Sawmill を本年 9 月 23 日 - 25 日、カリフォルニア州サンノゼで開催される Streaming Media West に出展します。今年のテーマは“オンラインビデオのビジネスと技術”です。当該期間、サンノゼ滞在予定のお持ちであれば、ご見学ください。事前にご連絡いただけましたら、Flowerfire 社にその旨申し送りします。

## 技術情報：Sawmill SQL を使用したログデータクエリー

Sawmill Enterprise はバックエンドデータベースとして MySQL サーバーをサポートします (これは Enterprise のみです)。Sawmill 独自の内蔵 DB は MySQL より高速であり、Sawmill の標準レポートを生成する場合は、通常こちらを推奨します。しかし MySQL には SQL クエリーを実行するという内蔵 BD に無い長所があります。このニュースレターでは Sawmill が生成した MySQL DB から SQL を使用して情報を抽出する方法について説明します。

### MySQL プロファイルとデータベース生成

MySQL をバックエンドデータベースとして使用するためには内蔵 DB を生成する場合と同じようにプロファイルを生成します。しかし Create Profile ウィザードで“Use MySQL Database”を選択します (繰り返しますが Enterprise だけです)。つぎに MySQL DB のホスト名、ユーザー名、パスワードを入力します。MySQL 通信にソケットを使用しており、ソケットが DB と同じロケーションに無い場合は、パス名の入力が必要です。最後に、プロファイル名と異なる DB 名 (スキーム) が必要であれば、異なる DB 名を入力します。



プロフィール生成を継続すると、最後に Sawmill DB が生成されます。Sawmill によるレポート閲覧をする必要はありません。DB を外部クエリーのためにのみ使用するのであれば、“build database”操作で MySQL DB に全てのテーブルを蓄積できます。“build database”が完了すると、Sawmill は分析、規格化し、全てのログデータを、MySQL DB に挿入します。そして関連項目テーブルを構築します。

### メインテーブルと項目テーブルのクエリー

DB のメインテーブルは“logfile”と呼ばれ、ログデータの各イベントが 1 列になります。この例では、5000 行の Apache ログファイルを解析しますので、ログデータの各行は異なるイベントです。そのため logfile は 5000 行（全ての例は mysql コマンドラインプログラムのキャプチャ）です。

( コマンド `mysql > select count (*) from logfile;`  )



Logfile の 1 列は :

( コマンド `mysql > select * from logfile limit 1;`  )

```
mysql> select * from logfile limit 1;
```

logfileid	date_time	hostnameitemnum	day_of_week	hour_of_day	hit_type	page	size_type	VIEW	screen_dimensions	screen_depth	hostname	id
1	1999-04-07 16:53:04	2	2	2	2	2	2	2	2	2	140.177.203.25	2

ログデータの 1 行目はこのようになります。

```
140.177.203.25 -- [07/Apr/1999:16:53:04 -0500] "GET / HTTP/1.0" 200 734 "-" Mozilla/4.04 [en] (X11; I; SunOS 5.6 sun4u)"
```

Date\_time はログデータに一致し、Size 欄の 734 はログデータの 734 に一致します。しかしそれ以外の欄は明瞭ではありません。その理由是非数値フィールドがログファイルでは“規格化”されているためです。それらの値は logfile に直接含まれず、補助テーブル (itemnum テーブル) に含まれ、これらの値の参照値が logfile に含まれます。例えば、ホスト名は logfile では 2 であり、hostname itemnum テーブル (テーブルは hostnameitemnum であり、itemnum と hostname 欄を含みます。それらは itemnum から hostname へのマップやその逆に使用されます) の 140.177.203.25 に相当します。SQL の hostnameitemnum のクエリーで、itemnum=2 の欄だけが選択され、その関係を示します。

( コマンド mysql > select \* from hostnameitemnum where itemnum = 2; )

```
mysql> select * from hostnameitemnum where itemnum = 2;
```

itemnum	hostname
2	140.177.203.25

### メインテーブル(logfile)を Itemnums テーブルに結合

1 つ以上の itemnum テーブルをメインテーブルに結合すると、Sawmill 独自レポートで表示されるものと同様な結果が得られます。例えば、“Top 10 hostnames”レポートを生成します。ログファイルの選択と合計で行います。hostname のグループ分けを行い、hostnameitemnum に結合し実在 hostname 取得すると、それが結果 (規格化された hostname itemnum) ですので、降順に並べ替え、上位 10 で打ち切ります。

( コマンド mysql > select i.hostname, sum(hits) as hits, sum(page\_views) as page\_views, sum(size) as size from logfile 1 left join hostnameitemnum i on i.hostname = i.itemnum group by hostname order by hits desc limit 10; )

```
mysql> select i.hostname, sum(hits) as hits, sum(page_views) as page_views, sum(size) as size from logfile 1 left join hostnameitemnum i on i.hostname = i.itemnum group by hostname order by hits desc limit 10;
```

hostname	hits	page_views	size
129.17.19.120	490	44	870729
192.17.19.148	515	59	2627870
140.177.203.25	308	70	8299036
129.17.19.120	240	50	1337940
192.17.19.148	270	29	423709
140.177.203.25	87	21	378402
140.177.203.25	78	29	876470
140.177.203.25	78	24	876470
140.177.203.25	78	9	438235

どんな Sawmill 標準レポートでも SQL で同様に生成可能です。

## Itemnum によるフィルタ

結果をフィルタすることを考えます。全てのフィルタータイプは SQL クエリーの WHERE の使用で可能です。この場合、.com hostname でフィルター表示します。”where i hostname like ‘%.com’”を含むと、クエリーはホスト名が.com で終わる値を含むものだけを選択します。

( コマンド `mysql > select i.hostname, su(hits) as hits, sum(page_view) as page_view, sum(size) as size from logfile 1 left join hostnameitemnum i on 1.hostname = i.itemnum where i.hostname like ‘%.com’ group by hostname order by hits desc limit 10; )`

```
mysql> select i.hostname, sum(hits) as hits, sum(page_views) as page_views, sum(size) as size from logfile 1 left join hostnameitemnum i on 1.hostname = i.itemnum where i.hostname like ‘%.com’ group by hostname order by hits desc limit 10;
```

hostname	hits	page_views	size
gala-hai.com	100	70	2100000
g11232.ctea.com	75	9	439318
gal.com	65	22	74708
tsakana.com	45	22	101878
tsarvika.com	39	6	104114
121232.ctea.com	34	6	10000
3-000-217-100-100.net	22	6	177808
tsakvika.com	22	6	102206
ts.com	15	1	10000
tsakvika.com	15	4	10000

フィルターはクエリー (hostname) の 1 次フィールドだけに有効ではありません。それは logfile の任意のフィールドあるいは他のフィールドの結合フィールドでも使用できます。MySQL は一つのクエリーでの多数の結合をサポートしているため、一つのテーブルでメインカラム(hostname)を結合し、また追加テーブルを結合してフィルタできます。たとえば、GIF イメージをアクセスしたホスト名( Hits 欄は書くホスト名による GIF アクセス数、Size 欄は各ホスト名アクセスによる転送数 ) です。

( コマンド `mysql > select i.hostname, sum(hits) as hits, sum(page_views) as page_views, sum(size) as size from logfile 1 left join hostnameitemnum i on 1.hostname = i.itemnum left join file_typeitemnum fi on 1.file_type = fi.itemnum where fi.file_type = ‘GIF’ group by hostname order by hits desc limit 10; )`

```
mysql> select i.hostname, sum(hits) as hits, sum(page_views) as page_views, sum(size) as size from logfile 1 left join hostnameitemnum i on 1.hostname = i.itemnum left join file_typeitemnum fi on 1.file_type = fi.itemnum where fi.file_type = ‘GIF’ group by hostname order by hits desc limit 10;
```

hostname	hits	page_views	size
1123232.ctea.com	45	0	447211
202.17.10.100	42	0	200000
ts.com	22	0	10000
102.17.10.100	15	0	101118
1123232.ctea.com	15	0	10000
1123232.ctea.com	15	0	432000
102.17.10.100	15	0	10000
102.17.10.100	15	0	10000
102.17.10.100	15	0	10000
102.17.10.100	15	0	10000

## 結論

このニュースターでは Sawmill の MySQL プロファイル生成と、それを使用したログデータの SQL DB へのインポートを説明しました。このプロセスで任意のログデータを MySQL DB にインポートできます。Sawmill を使って、( 1 ) バックエンド DB として MySQL を使用して Sawmill プロファイルを作成、( 2 ) DB を構築、( 3 ) SQL クエリーを実行、の手順でどのようなログデータに対しても任意の SQL クエリーを実行することが

可能です。