

中央研究院 資訊科技創新研究中心

OpenFoundry 計畫報告

開發人員：尤正偉 <cwyu.cs@gmail.com>

專案經理：黃崇閔<charles.huang@citi.sinica.edu.tw>

執行期間：102 年 12 月 12 日至 103 年 02 月 28 日

摘要

Subversion (簡稱 SVN) 是一套成熟及穩定的版本控制系統，簡單易用的特性成為許多程式開發者使用的工具。對於一個使用 SVN 的軟體專案而言，往往只有曾經參與其中的開發者了解該專案的歷史背景，其他人只能夠從文字記錄檔中，個別檢視不同版本資訊的歷史紀錄。我們使用 SVNPlot (Generate graphs from Subversion log statistics) 這套 Open Source 工具及線上專案管理平台 (Google Code, SourceForge, OpenFoundry) 當成資料來源，把記錄檔整合到資料庫，並可即時更新資料，讓使用者可以用視覺化的圖形快速了解軟體專案的開發情形與活躍程度，例如：專案的開發時間曲線圖、開發者貢獻比較圖、開發者所有參與的歷史專案列表等。

目的、需求

程式需要動態地產生視覺化的圖形，並可在網頁上呈現，當 Subversion 專案有更新的時候，必須要能夠即時同步。

相關工作

產生視覺化的圖形之前我們必須先有資料，看有沒有可用的 Open Source 或是函式庫等工具可以直接應用，或是協助抓到 SVN log 資料。初步我們以 SVMModel [1]、phpsvnclient [2] 等做為抓資料的參考工具，然後去想如果拿到了所有專案的資料後，應該怎麼呈現給使用者 (人、事、物、時間等)，並同時去網路上搜尋看看是不是已經有人做過類似的工作，或是有沒有現成可用的視覺化工具，找到了包括這些：WebSVN [3]、GitHub user UI [4]、jqPlot [5]、Gource [6]、StatSVN [7]、SVNPlot [8]、Highcharts JS [9] 等。

其中以 Gource 呈現效果最為華麗，但是是以影片方式呈現畫面。StatSVN 可以利用 SVN log 分析出單一專案的開發人員有幾位、各自做了多少事情、寫了幾行 Code、目錄大小、專案行數等分析結果，並產生可顯示在網頁上的 JavaScript 圖形。SVNPlot 可以去抓 SVN log，產生 SQLite 資料庫檔案 (local

file)，再利用 jqPlot 產生專案的 JavaScript 圖形。Highcharts JS 是專門用來處理數據圖表的視覺化工具，呈現效果佳，只要把資料餵進去，就可以幫你產生出 JavaScript 圖形。

程式實作

我們以 OpenFoundry 的 Subversion 專案為例，使用 SVNPlot 開放原始碼 (Linux version)，定期地將每個專案的 log 抓下來，個別產生 *.sqlite 資料庫檔案、和個別的 JavaScript 圖形 (option，可在設定檔 config.xml 中設定)，再利用 sqlite3-to-mysql.py 這個開放原始碼工具將 *.sqlite 轉換成 MySQL 語法，然後再將資料整合到 MySQL 的同一個 database 中。並設計立即更新的功能 (update_now)，讓使用者可為單一專案立即更新資料。至於所有專案整體的視覺化圖形呈現部分，則留至 Feature Work。

此程式使用 PHP + Shell Script 完成。PHP 用來提供 HTTP Service 並啟動 Shell Script 執行 SVNPlot 說明文件中所需指令。

PHP 程式碼片段

```
// scan all svn repositories, and execute program in background respectively
foreach ($svn_list as $svn_repository) {
    // check if data is up-to-date
    if (isUpToDate($svn_repository) == true) {
        // output message
        echo "<tr><td>" . $svn_repository . "</td><td>" . "revision up-to-date, do nothing" . "</td></tr>";
    } else {
        // variables
        $script = $config->fetch_sqlite_to_mysql_script;

        // get running job(s) if any
        $pid_info = shell_exec("pgrep -fl \"\$script \$svn_repository\" | grep -v pgrep");

        // set log command (redirect stdout/stderr to log file)
        $log_redirect = ($config->is_logging == "yes") ? "&> $config->log_file" : "";

        // fetch data in normal priority
        if (empty($pid_info)) {
            // execute program (background execution)
            pclose(popen("sh && \$script \$svn_repository \$log_redirect &", "w")); // "sh &&" for two-level process

            // output message
            echo "<tr><td>" . $svn_repository . "</td><td>" . "revision out-of-date, now fetching" . "</td></tr>";
        } else {
            // output message
            echo "<tr><td>" . $svn_repository . "</td><td>" . "revision out-of-date, already running" . "</td></tr>";
        }
    }

    // flush stdout
    ob_flush();
    flush();
}
```

上圖程式片段會用迴圈一個個地去檢查每個專案是否該更新了，儘管 SVNPlot 會自己判定目前抓到哪個版本，接下來應該要抓哪些剩下的部份，但我們為了效率的考量，讓 SVNPlot 在判定之前，先行檢查，如果某個專案的版本已經是最新的，就不去執行 SVNPlot，跳到迴圈下一個專案去。如果資料是需要更新的時候，就利用 pclose(pcopen("指令")); 這個 non-blocking 的函式呼叫去背景執行 SVNPlot 相關的 Shell Script。圖中 isUpToDate() 函式是根據 <

<http://svn.openfoundry.org/專案名稱> 這個由 OpenFoundry 提供的網站版本資訊，和上一次 SVNPlot 執行完之後留下的版本資訊，兩者比對的結果。

Shell Script 程式碼片段

```
#####
# DESCRIPTION:
#   (Use SVNplot Toolkit)
#   Fetch subversion repository log data into SQLite local file
#
# IN:
#   $1    project name
#
# RETURNS:
#   NONE
#
# REFERENCE:
#   https://code.google.com/p/svnplot/
#####
function fetch_sqlite()
{
    # parameter
    repository=$1

    #### # svnplot fetch database (using http:// protocol)
    #### python /usr/lib/python2.6/site-packages/svnplot svnlog2sqlite.py -l -v http://svn.openfoundry.org/$repository/ $output_path/$repository.sqlite

    # SVNPlot fetch svn log data into SQLite database (using file:// protocol)
    svnadmin create $output_path$mirrors/$repository
    cat /dev/null > $output_path$mirrors/$repository/hooks/pre-revprop-change
    echo "#!/bin/sh" >> $output_path$mirrors/$repository/hooks/pre-revprop-change
    echo 'exit 0;' >> $output_path$mirrors/$repository/hooks/pre-revprop-change
    chmod +x $output_path$mirrors/$repository/hooks/pre-revprop-change
    svnsync initialize file://$output_path$mirrors/$repository http://svn.openfoundry.org/$repository/
    svnsync synchronize file://$output_path$mirrors/$repository
    python /usr/lib/python2.6/site-packages/svnplot svnlog2sqlite.py -l -v file://$output_path$mirrors/$repository $output_path/$repository.sqlite

    # SVNPlot generate graphic visualization
    if [ "$is_graph" == "yes" ]; then
        mkdir -p $output_path$svnplot$repository
        python -m svnplot.svnplot-js -v $output_path/$repository.sqlite $output_path/$repository
    fi
}
```

上圖程式片段主要是依照 SVNPlot 官方的說明文件做成，可以把單一專的 SVN log 抓成一個 .sqlite 檔案，並繪製成 JavaScript 圖形。官方提供兩種方式，一種是抓資料的過程中不在本機端暫存資料 (http:// protocol，上圖中註解起來的部分)，另一種則需要暫存資料 (file:// protocol)。前者抓資料的速度慢很多，抓完全部的專案大概需要 2 天時間；我們用的是後者，大概幾小時內就可以從無到有全部抓完。

Update Now 相關程式碼片段

```
// get stop command
void get_stop_command(char *command, const char *pid_list)
{
    // system command `pkill` with parent pid-list options
    sprintf(command, "pkill -P %s\\0", pid_list);
}

// get start command
void get_start_command(char *command, const char *script, const char *svn_repository)
{
    // system command `nice` with priority options
    sprintf(command, "nice -n %d %s %s\\0", PRIORITY, script, svn_repository);
}
```

上圖程式片段是用寫成的立即更新功能的相關程式碼，想像如果當下有定期排程已經在執行了，那麼我們就先將該專案的工作停止 (使用 Linux 指令 `pkill`)，再用較高的優先權 (使用 Linux 指令 `nice`) 執行一個新的。然而因為 Linux 系統上的考量，要給最高優先權的 nice 值需要超級使用者 (root) 的權

限，所以我們需要讓這段提高優先權的程式碼執行的瞬間變成 root，於是需要設定其 Effective User ID，但是因為 Linux 系統安全上的考量 Shell Script 無法設定 Effective User ID，所以才將此程式設計成 C 語言 (set-priority.c)，並在 Makefile 中編譯和設定 Effective User ID。

修改 SVNPlot 原始碼 (svnlog2sqlite.py)

```
# Before
451 def CreateTables(self):
452     cur = self.dbcon.cursor()
453     cur.execute("create table if not exists SVNLog(revno integer, commitdate timestamp, author text, msg text, \
454                 addedfiles integer, changedfiles integer, deletedfiles integer)")
455     cur.execute("create table if not exists SVNLogDetail(revno integer, changedpathid integer, changetype text, copyfrompathid integer, copyfromrev integer, \
456                 pathtype text, linesadded integer, linesdeleted integer, lc_updated char, entrytype char)")
457     cur.execute("CREATE TABLE IF NOT EXISTS SVNPaths(id INTEGER PRIMARY KEY AUTOINCREMENT, path text, relpathid INTEGER DEFAULT null)")

# After (add parameter "id INTEGER PRIMARY KEY AUTOINCREMENT")
451 def CreateTables(self):
452     cur = self.dbcon.cursor()
453     cur.execute("create table if not exists SVNLog(revno integer, commitdate timestamp, author text, msg text, \
454                 addedfiles integer, changedfiles integer, deletedfiles integer, id INTEGER PRIMARY KEY AUTOINCREMENT)")
455     cur.execute("create table if not exists SVNLogDetail(revno integer, changedpathid integer, changetype text, copyfrompathid integer, copyfromrev integer, \
456                 pathtype text, linesadded integer, linesdeleted integer, lc_updated char, entrytype char, id INTEGER PRIMARY KEY AUTOINCREMENT)")
457     cur.execute("CREATE TABLE IF NOT EXISTS SVNPaths(id INTEGER PRIMARY KEY AUTOINCREMENT, path text, relpathid INTEGER DEFAULT null)")


```

SVNPlot 會在 SQLite 裡面分別產生 SVNLog, SVNLogDetail 和 SVNPaths 這 3 個 Table，其中只有 SVNPaths Table 有 primary key，因為我們考量到如果只將資料單純地 insert 到 MySQL 而不做資料重複檢查的時候，在某些情況下可能會有資料重複的問題 (例如刪掉 .sqlite 使程式認定為新專案)，所以我們是在 insert 資料到 MySQL 的時候是用 REPLACE INTO 這個可以檢查資料重複的語法，如下圖所示 (fetch-sqlite-to-mysql.sh)。

```
# append data into target database (heredoc syntax)
mysql -u$db_user -p$db_password -h$db_hostname <<-EOF
    replace into `'$db_target`.'`$table_1`
        select * from `'$db_name`.'`$table_1`;
    replace into `'$db_target`.'`$table_2`
        select * from `'$db_name`.'`$table_2`;
    replace into `'$db_target`.'`$table_3`
        select * from `'$db_name`.'`$table_3`;
>---EOF
```

註：圖中底下 EOF 左方的 >--- 符號是 TAB 字元，是 Bash shell 允許 heredoc 可以縮排的語法。

如此，需要每個 Table 都加上 primary key，但是與其在產生出來的 .sqlite 中過濾，我們選擇直接修改 SVNPlot 原始程式碼：在 SVNLog 和 SVNLogDetail 建立 Table 的副程式參數尾端加入 id 這個自動變數，即 “id INTEGER PRIMARY KEY AUTOINCREMENT”，因為是加在尾端，Python 在呼叫的時候參數不足會給 null，不會 error，接著再在 Shell Script 中加入此變數 id 和 project 這個區分所有專案的新欄位當成 primary key，如下圖所示

(fetch-sqlite-to-mysql.sh)。

```
alter table `'$db_name`.'`$table_1` drop primary key, add primary key (`'$new_field`', `id`);  
alter table `'$db_name`.'`$table_2` drop primary key, add primary key (`'$new_field`', `id`);  
alter table `'$db_name`.'`$table_3` drop primary key, add primary key (`'$new_field`', `id`);
```

產生出來的 MySQL primary key 如下圖所示。

SVNLog Table: primary key (project, id)

#	名字	類型	排序規則	屬性	空	預設	額外
1	<u>project</u>	char(100)	utf8_general_ci		否	database-skeleton	
2	<u>id</u>	int(11)			否	0	
3	<u>revno</u>	int(11)			是	NULL	
4	<u>commitdate</u>	timestamp		on update CURRENT_TIMESTAMP	否	CURRENT_TIMESTAMP ON UPDATE	
5	<u>author</u>	text	utf8_general_ci		是	NULL	
6	<u>msg</u>	text	utf8_general_ci		是	NULL	
7	<u>addedfiles</u>	int(11)			是	NULL	
8	<u>changedfiles</u>	int(11)			是	NULL	
9	<u>deletedfiles</u>	int(11)			是	NULL	

SVNLogDetail Table: primary key (project, id)

#	名字	類型	排序規則	屬性	空	預設	額外	動作
1	<u>project</u>	char(100)	utf8_general_ci		否	database-skeleton		
2	<u>id</u>	int(11)			否	0		
3	<u>revno</u>	int(11)			是	NULL		
4	<u>changedpathid</u>	int(11)			是	NULL		
5	<u>changetype</u>	text	utf8_general_ci		是	NULL		
6	<u>copyfrompathid</u>	int(11)			是	NULL		
7	<u>copyfromrev</u>	int(11)			是	NULL		
8	<u>pathtype</u>	text	utf8_general_ci		是	NULL		
9	<u>linesadded</u>	int(11)			是	NULL		
10	<u>linesdeleted</u>	int(11)			是	NULL		
11	<u>lc_updated</u>	char(1)	utf8_general_ci		是	NULL		
12	<u>entrytype</u>	char(1)	utf8_general_ci		是	NULL		

SVNPaths Table: primary key (project, id)

#	名字	類型	排序規則	屬性	空	預設	額外	動作
1	<u>project</u>	char(100)	utf8_general_ci		否	database-skeleton		
2	<u>id</u>	int(11)			否	0		
3	<u>path</u>	text	utf8_general_ci		是	NULL		
4	<u>relpathid</u>	int(11)			是	NULL		

程式為了效能，利用 Linux 的指令 `diff` 先行比對，再將差異的部分 insert 到 MySQL 資料庫中，所以除了 SVNPlot 產生出來的 .sqlite 之外，我們還自行拷貝了一份上次的資料並儲存為 .sqlite.old 檔案。而 .sqlite.old.revision 則是去撈 SVNPlot 使用 http:// protocol 產生出來的暫存檔中裡面儲存的版本號資訊，可來當做讓 parse-websvn.php 程式判定是否為最新版本的依據，如下圖所示。

```
-rw-r--r-- 1 cwyu cwyu 49152 Mar 21 04:06 a12345.sqlite
-rw-r--r-- 1 cwyu cwyu 49152 Mar 21 04:06 a12345.sqlite.old
-rw-rw-rw- 1 cwyu cwyu      3 Mar 21 04:06 a12345.sqlite.old.revision
```

程式使用說明

Step 1：設定檔 config.xml

```

<!-- MySQL Information -->
<mysql>
    <hostname description="or IP address">localhost</hostname>
    <username>cwyu</username>
    <password>openfoundry</password>
    <output>
        <database description="all SQLite repositories into one MySQL database">svnplot</database>
        <new-field description="distinguish repositories in one MySQL database">project</new-field>
    </output>
    <resource>
        <database-skeleton>
            <file>./resource/database-skeleton.sqlite</file>
        </database-skeleton>
    </resource>
</mysql>

<!-- SQLite Information -->
<sqlite>
    <output>
        <graph description="generate svnplot graph, yes or no">yes</graph>
        <directory>
            <database description="save *.sqlite files">database</database>
            <mirrors description="save repository clone copy in local for speed up">mirrors</mirrors>
            <svnplot description="save svnplot graphs">svnplot</svnplot>
        </directory>
    </output>
</sqlite>

<!-- Log Information -->
<log description="log message on 'fetch-sqlite-to-mysql.sh' script">
    <enable description="yes or no">yes</enable>
    <file description="write permission is required">./log.txt</file>
</log>

```

設定 MySQL 帳密、是否產生 JavaScript 圖形和是否記錄 log.txt (需注意檔案會逐漸變大) 等相關資訊。

Step 2：執行 make

```
# output directory for *.sqlite files
tests -d "xllint --nodata --shell config.xml <> `cat /config/sqlite/output/directory/database/text()`" | grep -v '^/' >|| mkdir -p `xllint --nodata --shell config.xml <> `cat /config/sqlite/output/directory/database/text()`` | grep -v '^/' >
# update_now service: compile executable file
tests -d "xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/directory/text()`" | grep -v '^/' >|| mkdir -p `xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/directory/text()`` | grep -v '^/' >
gcc xllint --nodata --shell config.xml <> `cat /config/program/set-priority/source-file/text()` | grep -v '^/' > -o xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/directory/text()` | grep -v '^/' > xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/filename/text()` | grep -v '^/' >
# update_now service: set the effective UID to root for others (ex. apache user)
sudo chown root:root xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/directory/text()` | grep -v '^/' >|| xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/directory/text()`` | grep -v '^/' > && sudo chmod u+o xllint --nodata --shell config.xml <> `cat /config/program/set-priority/executable/filename/text()` | grep -v '^/' >
#[udool password] <> `cat /config/udool/skeleton: minimize queries
sqlite3 tmp --nodata --shell config.xml <> `cat /config/mysql/resource/database-skeleton/file/text()` | grep -v '^/' > .dump | grep -i "CREATE TABLE" | sqlite3 tmp & mv -f tmp xllint --nodata --shell config.xml <> `cat /config/mysql/resource/database-skeleton/file/text()` | grep -v '^/' >
# create database skeleton
./create-database-skeleton.sh
ERROR 1050 (42S01) at line 2: Table 'SVNLog' already exists
ERROR 1066 (42S21) at line 1: Duplicate column name 'project'
[udool password]
# set log file writable
touch `xllint --nodata --shell config.xml <> `cat /config/log/file/text()`` | grep -v '^/' > && chmod 666 `xllint --nodata --shell config.xml <> `cat /config/log/file/text()`` | grep -v '^/' >
phpwhois [var/www/html/code] -cyyu -l
```

設定、編譯相關程式 (過程中需超級使用者權限產生 `set-priority` 程式，可能需要輸入 `sudo` 密碼)，若 MySQL 沒有 `target database`，則會根據設定檔 (預設 `svnplot`) 產生一個空的，如下圖所示。

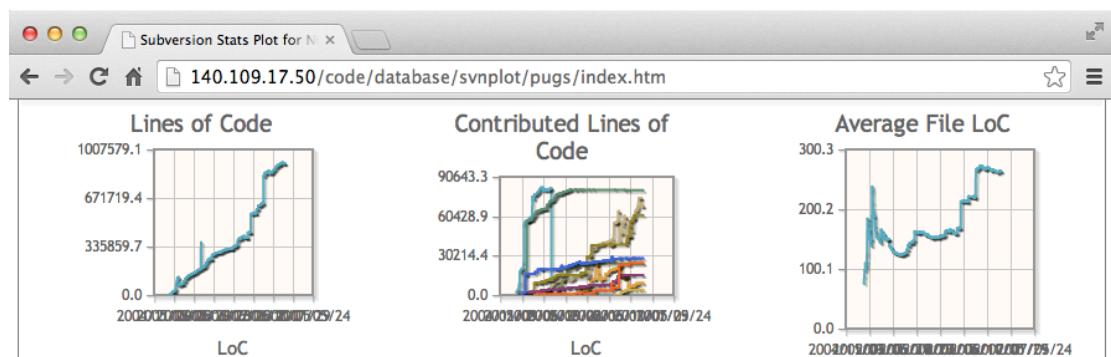
localhost » svnplot								
結構		SQL		搜尋		查詢		匯出
表		動作				操作		權限
<input type="checkbox"/>	SVNLog							0 MyISAM utf8_general_ci 2 KB -
<input type="checkbox"/>	SVNLLogDetail							0 MyISAM utf8_general_ci 2 KB -
<input type="checkbox"/>	SVNPaths							0 MyISAM utf8_general_ci 2 KB -
3 張資料表		總計						0 MyISAM utf8_general_ci 6 KB 0 B

Step 3：設定排程 (系統設定檔 /var/spool/cron/cwyu)

```
*/15 * * * * cd /var/www/html/code/ && php parse-websvn.php
```

參考設定：每 15 分鐘執行一次。

附註：產生的 JavaScript 圖形 (config.xml 可設定圖形是否產生)



參考範例：pugs 專案。

附註：產生的 local 端檔案相關位置

```
cwyu@phpsvn: ~ — ssh — 77x32
phpsvn [/var/www/html/code/database] -cwyu- ll -h a12345.sqlite*
-rw-r--r-- 1 cwyu cwyu 48K Mar 20 14:55 a12345.sqlite
-rw-r--r-- 1 cwyu cwyu 48K Mar 20 14:55 a12345.sqlite.old
-rw-rw-rw- 1 cwyu cwyu    3 Mar 20 14:53 a12345.sqlite.old.revision
phpsvn [/var/www/html/code/database] -cwyu- ll -h svnplot/a12345/
total 524K
-rw-r--r-- 1 cwyu cwyu 12K Mar 20 21:41 excanvas.compiled.js
-rw-rw-rw- 1 cwyu cwyu 25K Mar 20 21:41 index.htm
-rw-r--r-- 1 cwyu cwyu 13K Mar 20 21:41 jqplot.barRenderer.min.js
-rw-r--r-- 1 cwyu cwyu 11K Mar 20 21:41 jqplot.categoryAxisRenderer.min.js
-rw-r--r-- 1 cwyu cwyu 6.6K Mar 20 21:41 jqplot.dateAxisRenderer.min.js
-rw-r--r-- 1 cwyu cwyu 14K Mar 20 21:41 jqplot.pieRenderer.min.js
-rw-r--r-- 1 cwyu cwyu 343K Mar 20 21:41 jquery.jqplot.js
-rw-r--r-- 1 cwyu cwyu 3.2K Mar 20 21:41 jquery.jqplot.min.css
-rw-r--r-- 1 cwyu cwyu 84K Mar 20 21:41 jquery.min.js
phpsvn [/var/www/html/code/database] -cwyu- ll -h mshow.sqlite*
-rw-rw-rw- 1 root apache 8.3M Mar 20 15:49 mshow.sqlite
-rw-rw-rw- 1 root apache 8.3M Mar 20 15:49 mshow.sqlite.old
-rw-rw-rw- 1 root cwyu      2 Mar 20 15:19 mshow.sqlite.old.revision
phpsvn [/var/www/html/code/database] -cwyu- ll -h svnplot/mshow/
total 524K
-rw-r--r-- 1 root apache 12K Mar 20 15:49 excanvas.compiled.js
-rw-rw-rw- 1 root apache 25K Mar 20 23:15 index.htm
-rw-r--r-- 1 root apache 13K Mar 20 15:49 jqplot.barRenderer.min.js
-rw-r--r-- 1 root apache 11K Mar 20 15:49 jqplot.categoryAxisRenderer.min.js
-rw-r--r-- 1 root apache 6.6K Mar 20 15:49 jqplot.dateAxisRenderer.min.js
-rw-r--r-- 1 root apache 14K Mar 20 15:49 jqplot.pieRenderer.min.js
-rw-r--r-- 1 root apache 343K Mar 20 15:49 jquery.jqplot.js
-rw-r--r-- 1 root apache 3.2K Mar 20 15:49 jquery.jqplot.min.css
-rw-r--r-- 1 root apache 84K Mar 20 15:49 jquery.min.js
phpsvn [/var/www/html/code/database] -cwyu-
> 2 <                               cwyu@phpsvn
                                         2014/03/21  2:08 AM
```

參考範例：a12345 專案。

程式執行結果分析

我們用壓力測試，資料庫一開始都清空的情況下，定期每 1 分鐘去執行程式，設定產生 JavaScript 圖形和 log.txt 檔案。共計 1384 個專案，程式總共跑了約 9 小時。

```
cwyu@phpsvn: ~ — ssh — 72x26
phpsvn [var/www/html/code/database] -cwyu- ll *.sqlite | wc -l
1384
phpsvn [var/www/html/code/database] -cwyu- ll -tr *.sqlite | head
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 a1013542ac.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 a123456.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 acc.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 abcdefghij.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 adccctmpp.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 aguotest.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 ahbanalyzeraxi.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 ahbaxianalyzer.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 adss.sqlite
-rw-r--r-- 1 cwyu cwyu 12288 2014-03-20 14:53 aesds1.sqlite
phpsvn [var/www/html/code/database] -cwyu- ll -tr *.sqlite | tail
-rw-r--r-- 1 cwyu cwyu 1880064 2014-03-20 23:18 extwind.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-20 23:28 demoforossf.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-20 23:31 visual-osa.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-20 23:54 wikipediapi.sqlite
-rw-r--r-- 1 cwyu cwyu 31744 2014-03-21 00:00 ofmanual.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-21 00:02 musicbar.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-21 00:02 wfapdev.sqlite
-rw-r--r-- 1 cwyu cwyu 22528 2014-03-21 00:03 funslope.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-21 00:13 ruview.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 2014-03-21 00:15 gwp.sqlite
phpsvn [var/www/html/code/database] -cwyu-
> 2 <                               cwyu@phpsvn                         2014/03/21 1:05 AM
```

圖中檔案大小 (column 5) 為 12288 的都是空的專案，產生出來的 .sqlite 只是一些建立 Table 的語法，裡面都沒有資料，所以 MySQL 裡面是找不到這些專案的，以 a1013542ac 專案為例，網站上看不到任何資料，如下圖所示。



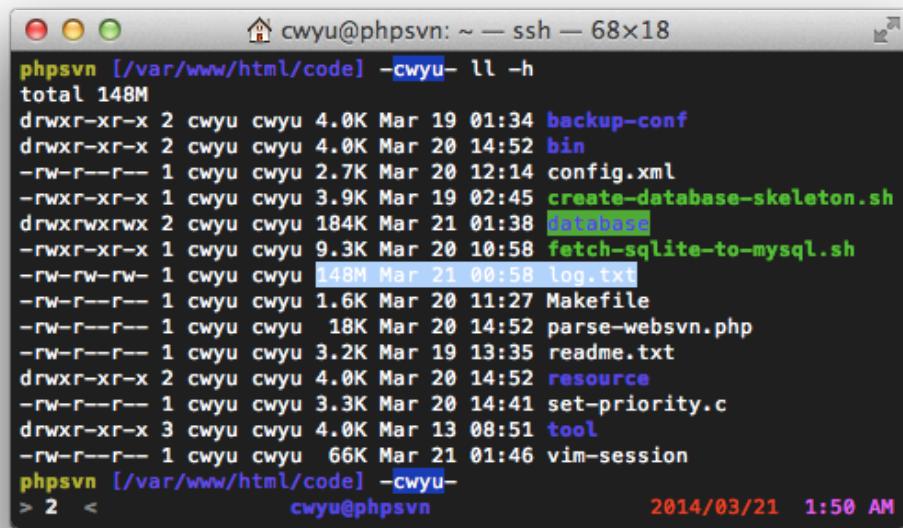
分析一下最後抓完 .sqlite 的時間，如下圖所示，可以看到 1384 個專案從開始 14:53PM 到 16:44PM 這段大約 2 個小時的期間，資料只剩下 50 個專案還沒抓完，這個部分可以留到 Feature Work 研究為什麼這些拖慢的專案為什麼需要抓那麼久 (建議看 log.txt 分析)。

```

cwyu@phpsvn: ~ — ssh — 71x53
phpsvn [/var/www/html/code/database] -cwyu- ll -tr *.sqlite | tail -n50
-rw-r--r-- 1 cwyu cwyu 6071296 Mar 20 16:44 utilvserver.sqlite
-rw-r--r-- 1 cwyu cwyu 602112 Mar 20 16:51 pixmicat.sqlite
-rw-r--r-- 1 cwyu cwyu 1700864 Mar 20 16:53 rt3elixus.sqlite
-rw-r--r-- 1 cwyu cwyu 7733248 Mar 20 16:57 joomla.sqlite
-rw-r--r-- 1 cwyu cwyu 1927168 Mar 20 16:57 swiss.sqlite
-rw-r--r-- 1 cwyu cwyu 1357824 Mar 20 16:57 dic.sqlite
-rw-r--r-- 1 cwyu cwyu 292864 Mar 20 16:58 idb.sqlite
-rw-r--r-- 1 cwyu cwyu 5604352 Mar 20 17:00 pcman.sqlite
-rw-r--r-- 1 cwyu cwyu 4884480 Mar 20 17:02 inuc20gui.sqlite
-rw-r--r-- 1 cwyu cwyu 1802240 Mar 20 17:06 jxosbsp.sqlite
-rw-r--r-- 1 cwyu cwyu 4307968 Mar 20 17:10 par.sqlite
-rw-r--r-- 1 cwyu cwyu 1768448 Mar 20 17:12 ttb.sqlite
-rw-r--r-- 1 cwyu cwyu 8904704 Mar 20 17:13 minidochun.sqlite
-rw-r--r-- 1 cwyu cwyu 8224768 Mar 20 17:16 logos.sqlite
-rw-r--r-- 1 cwyu cwyu 8460288 Mar 20 17:21 disirer2008.sqlite
-rw-r--r-- 1 cwyu cwyu 672768 Mar 20 17:21 eventlist.sqlite
-rw-r--r-- 1 cwyu cwyu 21636096 Mar 20 17:34 ossf.sqlite
-rw-r--r-- 1 cwyu cwyu 1566720 Mar 20 17:35 openapi.sqlite
-rw-r--r-- 1 cwyu cwyu 26361856 Mar 20 17:38 taibif.sqlite
-rw-r--r-- 1 cwyu cwyu 5822464 Mar 20 17:39 moztw.sqlite
-rw-r--r-- 1 cwyu cwyu 3903488 Mar 20 17:43 inuc.sqlite
-rw-r--r-- 1 cwyu cwyu 27279360 Mar 20 17:45 sahanapy.sqlite
-rw-r--r-- 1 cwyu cwyu 30911488 Mar 20 17:51 openfoundry.sqlite
-rw-r--r-- 1 cwyu cwyu 294912 Mar 20 18:36 nsc102mss.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 20:04 algorepu.sqlite
-rw-r--r-- 1 cwyu cwyu 9352192 Mar 20 20:23 pugs.sqlite
-rw-r--r-- 1 cwyu cwyu 229376 Mar 20 20:26 libchewingdata.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 22:38 paketest.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 22:43 happysource.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 22:54 visiondotnet.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 22:57 pythonext.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 22:58 facefoundry.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:00 wowarpprotector.sqlite
-rw-r--r-- 1 cwyu cwyu 36864 Mar 20 23:02 backupmanager.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:02 dbdmock.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:04 openbizsoft.sqlite
-rw-r--r-- 1 cwyu cwyu 115712 Mar 20 23:07 bamboo.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:08 delphiktoperp.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:17 efutility.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:17 unicloud2.sqlite
-rw-r--r-- 1 cwyu cwyu 1880064 Mar 20 23:18 extwind.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:28 demoforossf.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:31 visual-osa.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 20 23:54 wikipediapi.sqlite
-rw-r--r-- 1 cwyu cwyu 31744 Mar 21 00:00 ofmanual.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 21 00:02 musicbar.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 21 00:02 wfapdev.sqlite
-rw-r--r-- 1 cwyu cwyu 22528 Mar 21 00:03 funslope.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 21 00:13 ruview.sqlite
-rw-r--r-- 1 cwyu cwyu 18432 Mar 21 00:15 gwp.sqlite
phpsvn [/var/www/html/code/database] -cwyu-
> 2 <                               cwyu@phpsvn                         2014/03/21 1:32 AM

```

下圖可看到 log.txt 檔案會變很大 (註：log.txt 檔案寫入順序有可能不是依序，因為每個專案再抓的時候都是背景執行，可能作業系統排程 round-robin 一下，就被切掉了)，可能需要偶爾注意一下，或是去設定檔 config.xml 關閉 log。

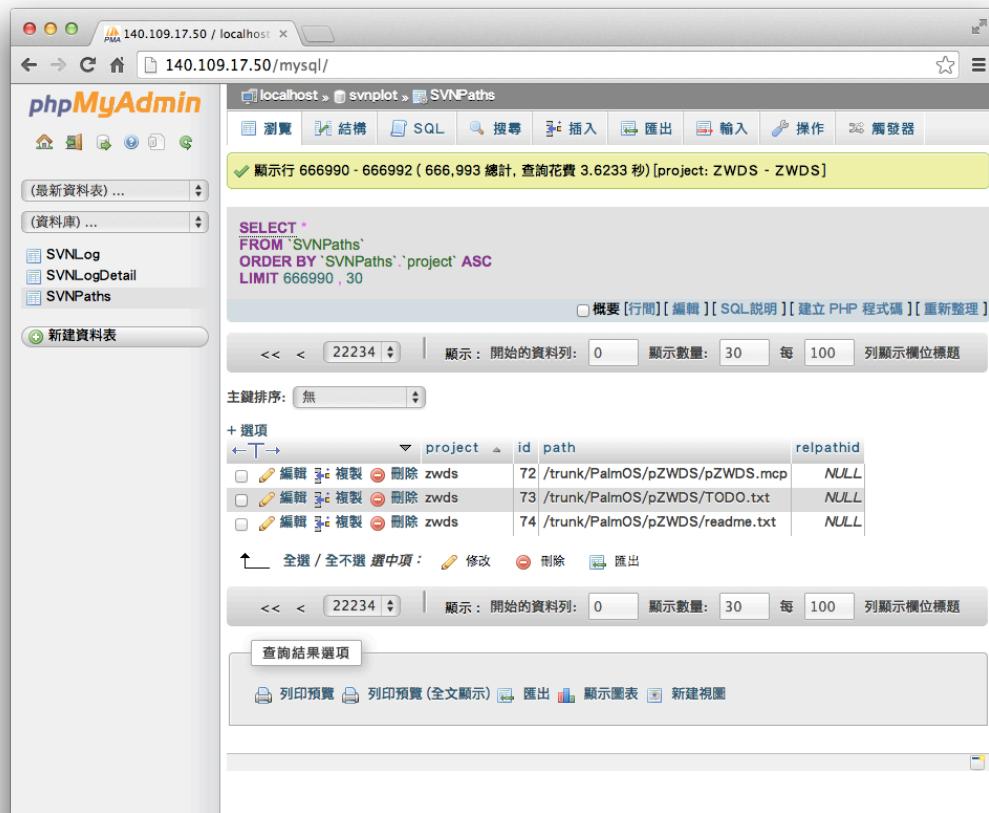


```

cwyu@phpsvn: ~ — ssh — 68x18
phpsvn [/var/www/html/code] -cwyu- ll -h
total 148M
drwxr-xr-x 2 cwyu cwyu 4.0K Mar 19 01:34 backup-conf
drwxr-xr-x 2 cwyu cwyu 4.0K Mar 20 14:52 bin
-rw-r--r-- 1 cwyu cwyu 2.7K Mar 20 12:14 config.xml
-rwxr-xr-x 1 cwyu cwyu 3.9K Mar 19 02:45 create-database-skeleton.sh
drwxrwxrwx 2 cwyu cwyu 184K Mar 21 01:38 database
-rwxr-xr-x 1 cwyu cwyu 9.3K Mar 20 10:58 fetch-sqlite-to-mysql.sh
-rw-rw-rw- 1 cwyu cwyu 148M Mar 21 00:58 log.txt
-rw-r--r-- 1 cwyu cwyu 1.6K Mar 20 11:27 Makefile
-rw-r--r-- 1 cwyu cwyu 18K Mar 20 14:52 parse-websvn.php
-rw-r--r-- 1 cwyu cwyu 3.2K Mar 19 13:35 readme.txt
drwxr-xr-x 2 cwyu cwyu 4.0K Mar 20 14:52 resource
-rw-r--r-- 1 cwyu cwyu 3.3K Mar 20 14:41 set-priority.c
drwxr-xr-x 3 cwyu cwyu 4.0K Mar 13 08:51 tool
-rw-r--r-- 1 cwyu cwyu 66K Mar 21 01:46 vim-session
phpsvn [/var/www/html/code] -cwyu-
> 2 < cwyu@phpsvn 2014/03/21 1:50 AM

```

下圖為 MySQL 的 SVNPaths Table 截圖。

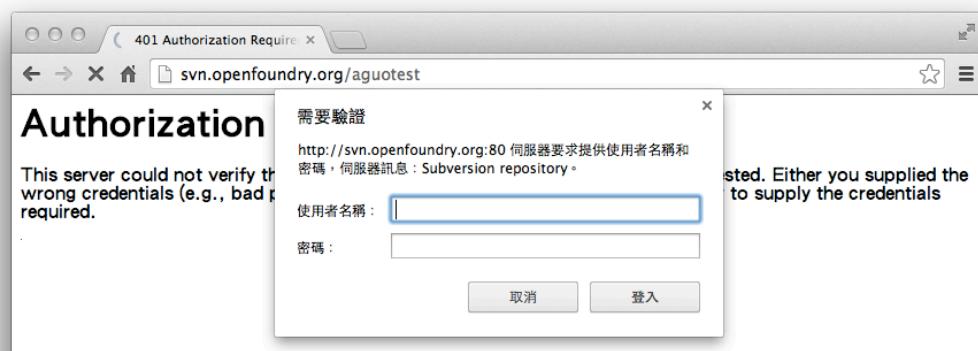


	project	id	path	relpathid
<input type="checkbox"/>	編輯 複製 刪除	72	/trunk/PalmOS/pZWDS/pZWDS.mcp	NULL
<input type="checkbox"/>	編輯 複製 刪除	73	/trunk/PalmOS/pZWDS/TODO.txt	NULL
<input type="checkbox"/>	編輯 複製 刪除	74	/trunk/PalmOS/pZWDS/readme.txt	NULL

當程式再排程執行一段時間後，用 HTTP Service 去執行全部專案，如下圖所示，可以發現，某些專案（圖中 aguotest, ahbanalyzeraxi, ahbaxianalyzer, ambaanalyzer 專案）不管執行幾次，就是會一直出現 “revision out-of-date, now fetching” 的訊息，永遠無法更新到最新狀態。

Project	Status
a1013542ac	revision up-to-date, do nothing
a12345	revision up-to-date, do nothing
a123456	revision up-to-date, do nothing
abcdefghij	revision up-to-date, do nothing
acc	revision up-to-date, do nothing
acrt	revision up-to-date, do nothing
ad3423453	revision up-to-date, do nothing
adcctrmpp	revision up-to-date, do nothing
ads7846	revision up-to-date, do nothing
adss	revision up-to-date, do nothing
aebs	revision up-to-date, do nothing
aesds1	revision up-to-date, do nothing
aesds2	revision up-to-date, do nothing
agdaexpcodes	revision up-to-date, do nothing
aguotest	revision out-of-date, now fetching
ahbanalyzeraxi	revision out-of-date, now fetching
ahbaxianalyzer	revision out-of-date, now fetching
ahmast	revision up-to-date, do nothing
ajaxerp	revision up-to-date, do nothing
ajaxperl	revision up-to-date, do nothing
ajaxshop	revision up-to-date, do nothing
alertfilter	revision up-to-date, do nothing
algorepu	revision up-to-date, do nothing
allab	revision up-to-date, do nothing
amazing	revision up-to-date, do nothing
ambaanalyzer	revision out-of-date, now fetching
ambatest	revision up-to-date, do nothing
amg	revision up-to-date, do nothing
amrcodec	revision up-to-date, do nothing

實地去看這些專案後發現，原來這些專案需要帳號密碼才能進去抓資料，Feature Work 可以考慮加入判定機制，讓遇到這些需要帳密的專案直接跳過。



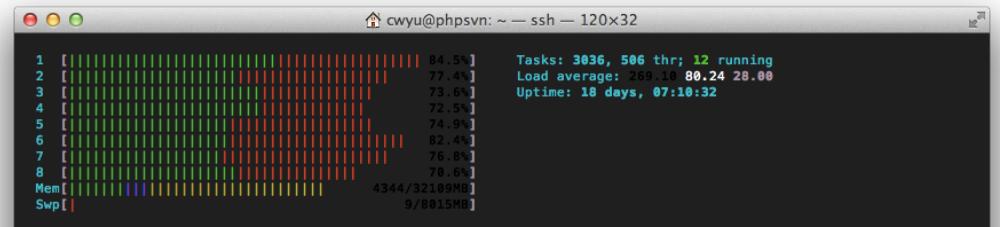
看系統 log 也可以看出大概是哪些專案需要帳密，如下圖所示。

開發過程：CPU Loading

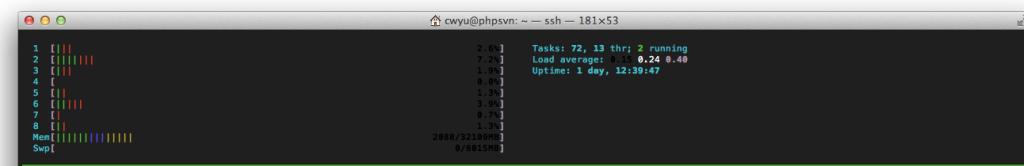
一開始資料全部都沒有的時候，執行程式，大概有上千個專案同時一起抓的情況下，CPU loading 會是程式的 bottleneck，如下圖所示。硬體資訊：CPU (2 core), Memory (8 GB)。



後來加強硬體，效能大幅改善，如下圖所示。硬體資訊：CPU (8 core), Memory (32 GB)。



確實，硬體加強可以改善 CPU 忙碌的問題，但是程式會需要如此多的資源是因為一開始資料全部都沒有的情況下同時去抓全部專案資料所造成。下圖為定期每 1 分鐘執行程式，而資料都已是最新的狀況。如果 Subversion 專案不是那麼常變動的話，或許之後可以考慮調整降低一下目前此提升後的硬體設備。



開發過程：svn checkout 無法共用

為了增進程式速度，想測試看看 SVNPlot 和用指令 `svn co 專案` 所產生的資料是否可以共用，如果相同且 `svn co 專案` 抓資料的速度又比較快的話，則考慮替換掉，但測試結果，不僅檔案目錄結構不同，就連檔案內容也不一樣，如下圖所示。

```
cwyu@phpsvn: ~ — ssh — 97x34
phpsvn [/var/www/html/code/database] -cwyu- ll
總計 36K
-rw-r--r-- 1 cwyu cwyu 1292 2014-03-04 03:33 a12345.sqlite-dump.by-svn-co.txt
-rw-r--r-- 1 cwyu cwyu 20619 2014-03-04 03:33 a12345.sqlite-dump.by-svnplot.txt
drwxr-xr-x 3 cwyu cwyu 4096 2014-03-04 03:37 by-svn-co
drwxr-xr-x 3 cwyu cwyu 4096 2014-03-04 03:28 by-svnplot
phpsvn [/var/www/html/code/database] -cwyu- ll by-svn-co/
總計 16K
-rw-r--r-- 1 apache apache 12288 2014-03-04 03:30 a12345.sqlite
drwxr-xr-x 3 cwyu cwyu 4096 2014-03-04 03:30 mirrors
phpsvn [/var/www/html/code/database] -cwyu- ll by-svn-co/mirrors/a12345/
總計 76K
drwxr-xr-x 6 cwyu cwyu 4096 2014-03-04 03:29 MultiMediaConverter
drwxr-xr-x 5 cwyu cwyu 4096 2014-03-04 03:29 NedNewServiceSetup
drwxr-xr-x 6 cwyu cwyu 4096 2014-03-04 03:29 .svn
drwxr-xr-x 6 cwyu cwyu 4096 2014-03-04 03:29 ThreadTimerDemo
-rw-r--r-- 1 cwyu cwyu 1957 2014-03-04 03:29 ThreadTimerDemo.sln
-rw-r--r-- 1 cwyu cwyu 49664 2014-03-04 03:29 ThreadTimerDemo.suo
drwxr-xr-x 7 cwyu cwyu 4096 2014-03-04 03:29 WindowsServiceDemo
phpsvn [/var/www/html/code/database] -cwyu- ll by-svnplot/
總計 40K
-rw-r--r-- 1 apache apache 33792 2014-03-04 03:27 a12345.sqlite
drwxr-xr-x 3 apache apache 4096 2014-03-04 03:27 mirrors
phpsvn [/var/www/html/code/database] -cwyu- ll by-svnplot/mirrors/a12345/
總計 24K
drwxr-xr-x 2 apache apache 4096 2014-03-04 03:27 conf
drwxr-sr-x 6 apache apache 4096 2014-03-04 03:27 db
-r--r--r-- 1 apache apache 2 2014-03-04 03:27 format
drwxr-xr-x 2 apache apache 4096 2014-03-04 03:27 hooks
drwxr-xr-x 2 apache apache 4096 2014-03-04 03:27 locks
-rw-r--r-- 1 apache apache 229 2014-03-04 03:27 README.txt
phpsvn [/var/www/html/code/database] -cwyu- vim -Rp a12345.sqlite-dump.by-svn*
```

> 2 <

cwyu@phpsvn

2014/03/04 3:43 AM

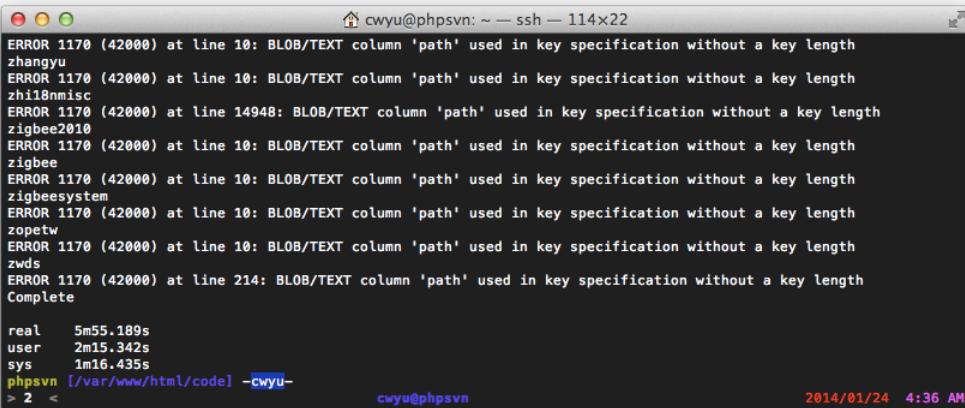
```
cwyu@phpsvn: ~ — ssh — 97x34
a12345.sqlite-dump.by-svn-co.txt a12345.sqlite-dump.by-svnplot.txt
1 PRAGMA foreign_keys=OFF;
2 BEGIN TRANSACTION;
3 CREATE TABLE SVNLog(revno integer, commitdate timestamp, author text, msg text,
4 CREATE TABLE SVNLogDetail(revno integer, changedpathid integer, changetype text, copyfrompath
5 CREATE TABLE SVNPaths(id INTEGER PRIMARY KEY AUTOINCREMENT, path text, relpathid INTEGER DEF
6 CREATE VIEW SVNLogDetailVw AS select SVNLogDetail.*, ChangedPaths.path as changedpath, CopyFr
7 CREATE INDEX svnlogrevnoidx ON SVNLog (revno ASC);
8 CREATE INDEX svnlogdtlrevnoidx ON SVNLogDetail (revno ASC);
9 CREATE INDEX svnlogdtlchangepathidx ON SVNLogDetail (changedpathid ASC);
10 CREATE INDEX svnlogdtlcopypathidx ON SVNLogDetail (copyfrompathid ASC);
11 CREATE INDEX svnpathidx ON SVNPaths (path ASC);
12 COMMIT;
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
~
N  RD a12345.sqlite-dump.by-svn-co.txt          unix | utf-8 | no ft   8% LN 1 C 1
> 2 <                                              cwyu@phpsvn                               2014/03/04 3:44 AM
```

```
cwyu@phpsvn: ~ — ssh — 97x34
a12345.sqlite-dump.by-svn-co.txt a12345.sqlite-dump.by-svnplot.txt
1 PRAGMA foreign_keys=OFF;
2 BEGIN TRANSACTION;
3 CREATE TABLE SVNLog(revno integer, commitdate timestamp, author text, msg text,
4 INSERT INTO "SVNLog" VALUES(1,'2008-07-20 16:10:47','ned','','20,0,0,1);
5 INSERT INTO "SVNLog" VALUES(2,'2008-07-20 16:23:23','ned','new check in',12,0,0,2);
6 INSERT INTO "SVNLog" VALUES(3,'2008-07-20 16:34:37','ned','','0,6,0,3);
7 INSERT INTO "SVNLog" VALUES(4,'2008-07-20 17:27:08','ned','','0,6,0,4);
8 INSERT INTO "SVNLog" VALUES(5,'2008-07-30 14:57:16','ned','check in encode by flix.',0,9,0,5)
9 INSERT INTO "SVNLog" VALUES(6,'2008-07-30 14:58:25','ned','check in encode demo by flix',24,0
10 INSERT INTO "SVNLog" VALUES(7,'2008-07-30 14:59:38','ned','delete file I don''t need.',0,0,2,
11 INSERT INTO "SVNLog" VALUES(8,'2008-08-02 14:28:50','ned','add config
12 add output message on window to display encoding process.',2,17,0,8);
13 INSERT INTO "SVNLog" VALUES(9,'2008-08-02 14:53:53','ned','整理Windows Service部分的程式碼',0
14 INSERT INTO "SVNLog" VALUES(10,'2008-08-17 17:23:29','ned','check in',0,15,0,10);
15 CREATE TABLE SVNLogDetail(revno integer, changedpathid integer, changetype text, copyfrompath
16 INSERT INTO "SVNLogDetail" VALUES(1,1,'A',NULL,NULL,'F',78,0,'Y','R',1);
17 INSERT INTO "SVNLogDetail" VALUES(1,2,'A',NULL,NULL,'D',0,0,'Y','R',2);
18 INSERT INTO "SVNLogDetail" VALUES(1,3,'A',NULL,NULL,'F',0,0,'Y','R',3);
19 INSERT INTO "SVNLogDetail" VALUES(1,4,'A',NULL,NULL,'F',0,0,'Y','R',4);
20 INSERT INTO "SVNLogDetail" VALUES(1,5,'A',NULL,NULL,'F',0,0,'Y','R',5);
21 INSERT INTO "SVNLogDetail" VALUES(1,6,'A',NULL,NULL,'F',752,0,'Y','R',6);
22 INSERT INTO "SVNLogDetail" VALUES(1,7,'A',NULL,NULL,'D',0,0,'Y','R',7);
23 INSERT INTO "SVNLogDetail" VALUES(1,8,'A',NULL,NULL,'D',0,0,'Y','R',8);
24 INSERT INTO "SVNLogDetail" VALUES(1,9,'A',NULL,NULL,'F',0,0,'Y','R',9);
25 INSERT INTO "SVNLogDetail" VALUES(1,10,'A',NULL,NULL,'D',0,0,'Y','R',10);
26 INSERT INTO "SVNLogDetail" VALUES(1,11,'A',NULL,NULL,'F',46,0,'Y','R',11);
27 INSERT INTO "SVNLogDetail" VALUES(1,12,'A',NULL,NULL,'F',9,0,'Y','R',12);
28 INSERT INTO "SVNLogDetail" VALUES(1,13,'A',NULL,NULL,'F',24,0,'Y','R',13);
29 INSERT INTO "SVNLogDetail" VALUES(1,14,'A',NULL,NULL,'D',0,0,'Y','R',14);
30 INSERT INTO "SVNLogDetail" VALUES(1,15,'A',NULL,NULL,'D',0,0,'Y','R',15);
N  RD a12345.sqlite-dump.by-svnplot.txt          unix | utf-8 | no ft   0% LN 1 C 1
> 2 <                                              cwyu@phpsvn                               2014/03/04 3:44 AM
```

開發過程：可忽略的錯誤訊息

在將資料 insert 到 MySQL 的過程中，log.txt 可以看到類似的錯誤訊息如下圖所示。

```
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
```



```
cwyu@phpsvn: ~ — ssh — 114x22
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
zhangyu
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
zhi18nmisc
ERROR 1170 (42000) at line 14948: BLOB/TEXT column 'path' used in key specification without a key length
zigbee2010
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
zigbee
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
zigbeesystem
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
zopetw
ERROR 1170 (42000) at line 10: BLOB/TEXT column 'path' used in key specification without a key length
zwds
ERROR 1170 (42000) at line 214: BLOB/TEXT column 'path' used in key specification without a key length
Complete

real    5m55.189s
user    2m15.342s
sys     1m16.435s
phpsvn [/var/www/html/code] -cwyu-                               cwyu@phpsvn
> 2 <                                         2014/01/24  4:36 AM
```

原因是 SQLite 轉到 MySQL 的時候，產生出來的 .sqlite 檔案中的參數 path 資料型態與 MySQL 不相容，如下圖所示。

```
CREATE_TABLE SVNPaths(id INTEGER PRIMARY KEY AUTOINCREMENT, path TEXT, relpathid INTEGER DEFAULT null);
```

手動改一下資料型態 (text -> varchar(100)) 就可以解決掉這個問題。

```
CREATE_TABLE SVNPaths(id INTEGER PRIMARY KEY AUTOINCREMENT, path varchar(100), relpathid INTEGER DEFAULT null);
```

網路參考資料 (<http://stackoverflow.com/questions/1827063/mysql-error-key-specification-without-a-key-length>)

5 Answers

active oldest votes

126

The error happens because MySQL can index only the first N chars of a BLOB or TEXT column. So The error mainly happen when there is a field/column type of TEXT or BLOB or those belongs to TEXT or BLOB types such as TINYBLOB, MEDIUMBLOB, LONGBLOB, TINYTEXT, MEDIUMTEXT, and LONGTEXT that you try to make as primary key or index. With full BLOB or TEXT without the length value, MySQL is unable to guarantee the uniqueness of the column as it's of variable and dynamic size. So, when using BLOB or TEXT types as index, the value of N must be supplied so that MySQL can determine the key length. However, MySQL doesn't support a key length limit on TEXT or BLOB. TEXT(88) simply won't work.

The error will also pop up when you try to convert a table column from non-TEXT and non-BLOB type such as VARCHAR and ENUM into TEXT or BLOB type, with the column already been defined as unique constraints or index. The Alter Table SQL command will fail.

The solution to the problem is to remove the TEXT or BLOB column from the index or unique constraint, or set another field as primary key. If you can't do that, and wanting to place a limit on the TEXT or BLOB column, try to use VARCHAR type and place a limit of length on it. By default, VARCHAR is limited to a maximum of 255 characters and its limit must be specified implicitly within a bracket right after its declaration, i.e VARCHAR(200) will limit it to 200 characters long only.

Sometimes, even though you don't use TEXT or BLOB related type in your table, the Error 1170 may also appear. It happens in situation such as when you specify VARCHAR column as primary key, but wrongly set its length or characters size. VARCHAR can only accepts up to 256 characters, so anything such as VARCHAR(512) will force MySQL to auto-convert the VARCHAR(512) to a SMALLTEXT datatype, which subsequently fail with error 1170 on key length if the column is used as primary key or unique or non-unique index. To solve this problem, specify a figure less than 256 as the size for VARCHAR field.

Reference: [MySQL Error 1170 \(42000\): BLOB/TEXT Column Used in Key Specification Without a Key Length](#)

[share](#) | [improve this answer](#)

answered Dec 1 '09 at 15:55

但因為實際上轉到 MySQL 的資料並沒有因為修改完而有差異，所以後來沒有去解決這個錯誤問題。

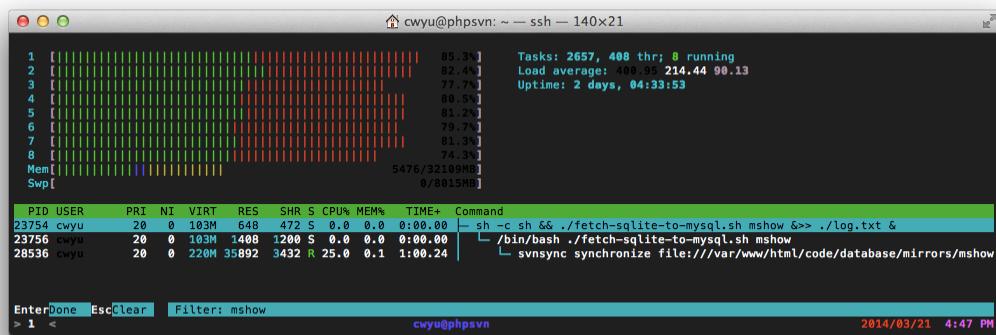
開發過程：檔案越抓越大

SVNPlot 兩個抓資料模式 (`http://` 和 `file://`)，我們採用 `file://` 這個抓資料速度比較快的模式，但是程式放著給它跑，過了一天、兩天，會發現暫存檔越來越大，測試過有到 75GB 的還在繼續增大就停掉程式不跑了。後來發現是 SVNPlot 程式的 bug [10]，裝了最新版本也沒有解決，所以就自己在我們的程式中想辦法解決，我們在 `fetch-sqlite-to-mysql.sh` 程式中執行完畢之前，都去將該次保留起來的暫存檔刪掉，因為我們有把暫存檔裡面的版本號資訊另外儲存起來，所以只有當真正 Subversion 專案有更新了，才會再重新讓 SVNPlot 去重抓資料，如下圖所示。

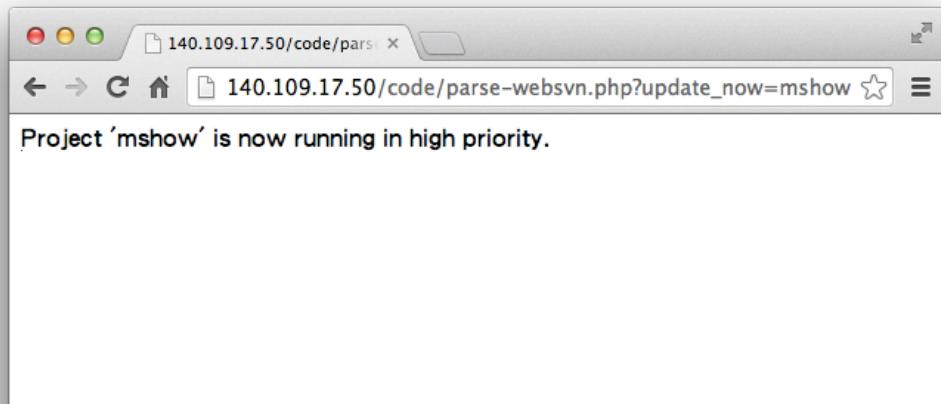
```
# avoid huge size (SVNPlot official bug <https://code.google.com/p/svnplot/issues/detail?id=57>)
rm -rf $output_path/$mirrors/$repository
echo "Complete"
```

開發過程：Update Now 功能

當排程定期在跑的時候，測試 Update Now 功能，以 mshow 專案為例。
原本的定期工作，NI (nice value) 值為 0 (nice value 範圍由低至高：19 ~ -20)，
如下圖所示。



使用 Update Now 功能 (HTTP Service)，如下圖所示。



結果原本的定期工作會被先停止掉，然後再生出一個新的工作，可以看到新工作的優先權 NI 值為較高的 -20。

```

Tasks: 2688, Load average: 42.11 245.08 105.69
Uptime: 2 days, 04:34:33
Mem: 5612/32109M
Swp[ 0/8015M]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
13159 root 20 0 3920 476 400 S 0.0 0.0 0:00.00 ./bin/set-priority --script=./fetch-sqlite-to-mysql.sh mshow --with-running-jobs
13188 root 0 -20 11304 1416 1192 S 0.0 0.0 0:00.00 /bin/bash ./fetch-sqlite-to-mysql.sh mshow
14766 root 0 -20 97764 4748 3168 S 0.0 0.0 0:00.04 svnsync synchronize file:///var/www/html/code/database/mirrors/mshow

EnterDone EscClear Filter: mshow
> 1 < cwyu@phpsvn 2014/03/21 4:48 PM

```

上述是目前程式碼的執行的結果，完全可以運作正常，只不過會多了一個 processss。

在執行 Update Now 功能的程式碼裡面，因為要多加呼叫 set-priority 程式碼來提高優先權，所以用系統程式去看 processss tree 會看到 3 層：

```
// first stop running jobs and then start new one (background execution)
pclose(popen("$set_priority --script=$script $svn_repository --with-running-jobs=$pid_list $log_redirect &", "w"));
```

level 1: PHP 呼叫 set-priority 程式

level 2: set-priority 呼叫 fetch-sqlite-to-mysql.sh 程式

level 3: 執行工作項目 (例如圖中的 svnsync)

而設定給定期執行的程式碼應該只有 2 層呼叫：

```
// execute program (background execution)
pclose(popen("$script $svn_repository $log_redirect &, "w"));
```

level 1: PHP 呼叫 fetch-sqlite-to-mysql.sh 程式

level 2: 執行工作項目 (例如 svnsync)

例如下圖這樣

```

Tasks: 510, Load average: 258.16 203.15
Uptime: 1 day, 01:36:23
Mem: 4216/32109M
Swp[ 0/8015M]

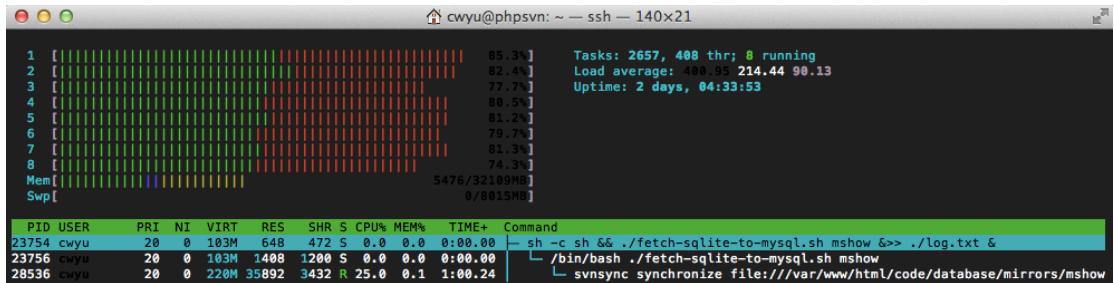
PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
16429 cwyu 20 0 103M 1488 1200 S 0.0 0.0 0:00.00 /bin/bash ./fetch-sqlite-to-mysql.sh mshow
24085 cwyu 20 0 228M 42640 3432 S 34.3 0.1 4:43.73 svnsync synchronize file:///var/www/html/code/database/mirrors/mshow

EnterDone EscClear Filter: mshow
> 1 < cwyu@phpsvn 2014/03/21 4:48 PM

```

可是當 Update Now 在執行的時候，會先砍掉上面這個定期執行的工作，然後再執行一個優先權較高的新的工作，但是在 2 層的架構下，新的工作卻每次都沒法執行起來，或是執行起來沒幾秒後就消失了，要再執行一次 Update Now 才會有新的工作開始正常執行。儘管後來也去檢查了相關的 process 有沒有誤

砍或是有沒有出現 zombie process，但始終找不出原因，後來就想到以前網路上看過的利用 double fork [11] 避免 zombie process 的概念，所以也用了 3 層的架構：



實作方式就是前面再多加一道 “sh &&” SHELL，讓它執行指令前先進入一個 SHELL 再執行指令：

```
// execute program (background execution)
pclose(popen("sh && $script $svn_repository $log_redirect &", "w"));
```

結果沒想到這樣就都正常了。我想應該還是我在砍掉定期工作的程式碼判定出了問題，或是砍掉了自己所造成，可以留到 Feature Work 研究，但是不解決也沒關係，因為目前程式可以正常運作，只是每個專案在抓的時候，系統資源就會多吃一個 process。實際測試整體效能，應該是沒有什麼影響。

Feature Work

1. MySQL -> Visualization Graph。從目前整合好的 MySQL 資料庫 (包含所有專案) 中抓資料，產生可在網頁上執行的視覺化動態圖形 (例如 JavaScript)。
2. 程式沒有考慮如果專案移除的部份，MySQL 資料庫要跟著同步的問題。
3. 上述『開發過程：Update Now 功能』中提到，double fork 改進。
4. 上述『程式執行結果分析』中提到，有些專案抓比較慢的原因。
5. 上述『程式執行結果分析』中提到，避開有些專案需要帳密才能抓資料的判定機制。

參考連結

- [1] SVMModel <http://www.php.net/manual/en/book.svn.php>
- [2] phpsvnclient <https://code.google.com/p/phpsvnclient/>
- [3] WebSVN <http://websvn.tigris.org/>
- [4] GitHub user UI <https://github.com/clkao>
- [5] jqPlot <http://www.jqplot.com/>
- [6] Gource <https://code.google.com/p/gource/>
- [7] StatSVN <http://www.statsvn.org/>
- [8] SVNPlot <https://code.google.com/p/svnplot/>
- [9] Highcharts JS <http://www.highcharts.com/>
- [10] SVNPlot Bug <https://code.google.com/p/svnplot/issues/detail?id=57>
- [11] Double Fork <http://thinkiii.blogspot.tw/2009/12/double-fork-to-avoid-zombie-process.html>