```
From b89aec0704290d5bae65c5bee72d196878179035 Mon Sep 17 00:00:00 2001

From: pomishra13 <pooja192009@gmail.com>

Date: Wed, 3 Dec 2014 16:23:00 -0800

Subject: [PATCH] news_updater patch


---

 bin/news_updater.php              |  1 +

 configs/config.php                |  7 +++-

 controllers/crawl_controller.php   | 65 +++++++++++++++++++++++++++++-

 controllers/machine_controller.php |  8 ++++

 models/machine_model.php          |  3 +-

 models/parallel_model.php         |  3 +-

 models/source_model.php           | 82 +++++++++++++++++++++++++-------------

 7 files changed, 137 insertions(+), 32 deletions(-)


diff --git a/bin/news_updater.php b/bin/news_updater.php

index 6048616..ed6da01 100644

--- a/bin/news_updater.php

+++ b/bin/news_updater.php

@@ -189,6 +189,7 @@ class NewsUpdater implements CrawlConstants

             $something_updated = true;

         }

         /*

+

             if anything changed rebuild shard

          */

         if($something_updated) {

diff --git a/configs/config.php b/configs/config.php

index 37f5def..35fcde1 100755
```

```
--- a/configs/config.php
+++ b/configs/config.php
@@ -728,4 +728,9 @@ define('NUM_FIELD_LEN', 4);
 define('WRITING_MODE_LEN', 5);
 /* Length of zero knowledge password string */
 define('ZKP_PASSWORD_LEN', 200);
-?>
\ No newline at end of file
+/**
+ *  set to true if Multiple news updaters are running
+ *  otherwise set to false if name server is running the news updater
+ */
+define('MULTIPLE_NEWS_UPDATER', true);
+?>
diff --git a/controllers/crawl_controller.php b/controllers/crawl_controller.php
index e6251da..1c837d0 100644
--- a/controllers/crawl_controller.php
+++ b/controllers/crawl_controller.php
@@ -59,8 +59,8 @@ class CrawlController extends Controller implements CrawlConstants
     var $activities = array("sendStartCrawlMessage", "sendStopCrawlMessage",
        "crawlStalled", "crawlStatus", "deleteCrawl", "injectUrlsCurrentCrawl",
        "getCrawlList", "combinedCrawlInfo", "getInfoTimestamp",
-       "getCrawlSeedInfo", "setCrawlSeedInfo", "getCrawlItems", "countWords",
-       "clearQuerySavePoint");
+       "getCrawlSeedInfo", "getNewsSources", "setCrawlSeedInfo", "getCrawlItems
+       ", "countWords","clearQuerySavePoint");
    /**
     * Checks that the request seems to be coming from a legitimate fetcher then
     * determines which activity the fetcher is requesting and calls that
```

```
@@ -322,5 +322,66 @@ class CrawlController extends Controller implements CrawlConstants
             TIMESTAMP_LEN);
         $this->model("crawl")->clearQuerySavePoint($save_timestamp);
     }
+    /**
+     * Handles the request to get the  array of news feed sources which hash to
+     * a particular value i.e. match with the index of requesting machine's
+     * hashed url/name from array of available machines hash
+     */
+     function getNewsSources()
+    {
+        if(!isset($_REQUEST["arg"])) {
+            return;
+        }
+        $current_machine = $this->clean(webdecode($_REQUEST["arg"]), "string");
+        $pre_feeds = $this->model("source")->getMediaSources("rss");
+        $pre_feeds = array_merge($pre_feeds,
+        $this->model("source")->getMediaSources("html"));
+        if(!$pre_feeds) { return false; }
+        $feeds = array();
+        foreach($pre_feeds as $pre_feed) {
+            if(!isset($pre_feed['NAME'])) {continue; }
+            $feeds[$pre_feed['NAME']] = $pre_feed;
+            if($pre_feed['TYPE'] == 'html') {
+                list($pre_feed['CHANNEL_PATH'], $pre_feed['ITEM_PATH'],
+                    $pre_feed['TITLE_PATH'], $pre_feed['DESCRIPTION_PATH'],
+                    $pre_feed['LINK_PATH']) =
+                    explode("###", html_entity_decode($pre_feed['AUX_INFO']));
+            }
+       }
+       $machine_urls = $this->model("source")->getMachineUrls();
+       $machine_array_length = count($machine_urls);
+       $feed_hash_values = array();
+       $i = 0;
+       foreach($feeds as $feed) {
+          if($feed) {
+             $hash = unpack( "N" ,(md5(substr($feed['NAME'], -2))));
+             $feed_hash_values[$i] = (($hash[1])%$machine_array_length);
+             $i++;
+          }
+       }
+       $i = 0;
+       $machine_index_match = 0;
+       foreach($machine_urls as $url) {
+          $url_hash = unpack( "N" ,(md5(substr($url['URL'], -2))));
+          $current_url_hash = ($url_hash[1]);
+          if(strcmp($current_url_hash, $current_machine ) == 0){
+             $machine_index_match = $i;
+             break;
+          }
+          $i++;
+       }
+       $hash_value = $feed_hash_values[$machine_index_match];
+       $news_sources = array();
+       $i = 0;
+       foreach($feeds as $feed) {
+          $feed_hash = unpack( "N" ,(md5(substr($feed['NAME'], -2))));
+          $current_feed_hash = (($feed_hash[1])%$machine_array_length);
+            if(strcmp($hash_value,$current_feed_hash) == 0){
+                $news_sources[$i] = $feed['NAME'];
+                $i++;
+            }
+        }
+        echo webencode(serialize($news_sources));
+    }
 }
 ?>
diff --git a/controllers/machine_controller.php b/controllers/machine_controller.php
index 47aee0d..abd03ed 100644
--- a/controllers/machine_controller.php
+++ b/controllers/machine_controller.php
@@ -82,6 +82,14 @@ class MachineController extends Controller implements CrawlConstants
      */
     function statuses()
     {
+        if(!isset($_REQUEST["arg"])) {
+                return;
+        }
+        $current_machine = $this->clean($_REQUEST["arg"], "string");
+        $machine_hash.= unpack( "N" ,(md5(substr($current_machine, -2))));
+        $hashValue =$machine_hash[1];
+        file_put_contents(WORK_DIRECTORY."/schedules/current_machine_info.txt",
+            $hashValue);
         echo json_encode(CrawlDaemon::statuses());
     }
     /**
diff --git a/models/machine_model.php b/models/machine_model.php
```

```
index 7ea8c3d..9368ba1 100644
--- a/models/machine_model.php
+++ b/models/machine_model.php
@@ -191,9 +191,10 @@ class MachineModel extends Model
         $time = time();
         $session = md5($time . AUTH_KEY);
         for($i = 0; $i < $num_machines; $i++) {
+            $url = $machines[$i]["URL"];
             $machines[$i][CrawlConstants::URL] =
                 $machines[$i]["URL"] ."?c=machine&a=statuses&time=$time".
-                "&session=$session";
+                "&session=$session&url=$url";
         }
         $statuses = FetchUrl::getPages($machines);
         for($i = 0; $i < $num_machines; $i++) {
diff --git a/models/parallel_model.php b/models/parallel_model.php
index 533e7d0..1e30676 100755
--- a/models/parallel_model.php
+++ b/models/parallel_model.php
@@ -477,6 +477,7 @@ class ParallelModel extends Model implements CrawlConstants
         $session = md5($time . AUTH_KEY);
         $query = "c=crawl&a=$command&time=$time&session=$session" .
             "&num=$num_machines";
+        crawlLog("the query is".$query);
         if($arg != NULL) {
             $arg = webencode($arg);
             $query .= "&arg=$arg";
@@ -485,7 +486,7 @@ class ParallelModel extends Model implements CrawlConstants
         $post_data = array();
         $i = 0;
         foreach($machine_urls as $index => $machine_url) {
-            $sites[$i][CrawlConstants::URL] =  $machine_url;
+            $sites[$i][CrawlConstants::URL] = $machine_url;
             $post_data[$i] = $query."&i=$index";
             $i++;
         }
diff --git a/models/source_model.php b/models/source_model.php
index 0274f6b..c53ec50 100644
--- a/models/source_model.php
+++ b/models/source_model.php
@@ -33,6 +33,8 @@
 if(!defined('BASE_DIR')) {echo "BAD REQUEST"; exit();}
 /** Loads the base class */
 require_once BASE_DIR."/models/model.php";
+/** Loads the ParallelModel class */
+require_once BASE_DIR."/models/parallel_model.php";
 /** IndexShards used to store feed indexes*/
 require_once BASE_DIR."/lib/index_shard.php";
 /** For text manipulation of feeds*/
@@ -45,7 +47,7 @@ require_once BASE_DIR."/lib/phrase_parser.php";
  * @package seek_quarry
  * @subpackage model
  */
-class SourceModel extends Model
+class SourceModel extends ParallelModel
 {
     /** Mamimum number of feeds to download in one try */
     const MAX_FEEDS_ONE_GO = 100;
```

```
@@ -123,6 +125,22 @@ class SourceModel extends Model
         return $row;
     }
     /**
+     * Receives a request to get machine data for an array of urls
+     */
+    function getMachineUrls()
+    {
+        $db = $this->db;
+        $machines = array();
+        $sql = "SELECT * FROM MACHINE";
+        $i = 0;
+        $result = $db->execute($sql);
+        while($machines[$i] = $db->fetchArray($result)) {
+            $i++;
+        }
+        unset($machines[$i]);
+        return $machines;
+            }
+    /**
      * Used to add a new video, rss, html news, or other sources to Yioop
      *
      * @param string $name
@@ -336,34 +354,44 @@ class SourceModel extends Model
      */
     function updateFeedItems($age = ONE_WEEK)
     {
-        $db = $this->db;
-        $time = time();
-       $feeds_one_go = self::MAX_FEEDS_ONE_GO;
-       $feeds = array();
-       $sql = "SELECT COUNT(*) AS CNT FROM MEDIA_SOURCE WHERE
-          TYPE='rss' OR TYPE='html'";
-       $result = $db->execute($sql);
-       $row = $db->fetchArray($result);
-       $num_feeds = (isset($row['CNT'])) ? $row['CNT'] : 0;
-       $num_bins = floor($num_feeds/$feeds_one_go) + 1;
-       $hour = date('H', $time);
-       $current_bin = $hour % $num_bins;
-       $limit = $current_bin * $feeds_one_go;
-       $limit = $db->limitOffset($limit, $feeds_one_go);
-       $sql = "SELECT * FROM MEDIA_SOURCE WHERE (TYPE='rss'
-           OR TYPE='html') $limit";
-       $result = $db->execute($sql);
-       $i = 0;
-       while($feeds[$i] = $this->db->fetchArray($result)) {
-          if($feeds[$i]['TYPE'] == 'html') {
-             list($feeds[$i]['CHANNEL_PATH'], $feeds[$i]['ITEM_PATH'],
+       if(MULTIPLE_NEWS_UPDATER) {
+          $current_machine = file_get_contents(WORK_DIRECTORY .
+             "/schedules/current_machine_info.txt");
+          $feeds = $this->execMachines("getNewsSources",array(NAME_SERVER),
+             $current_machine);
+          $result = @unserialize(webdecode($feeds[0][self::PAGE]));
+        } else {
+          $db = $this->db;
+          $time = time();
+          $feeds_one_go = self::MAX_FEEDS_ONE_GO;
+       $feeds = array();
+       $sql = "SELECT COUNT(*) AS CNT FROM MEDIA_SOURCE WHERE
+         TYPE='rss' OR TYPE='html'";
+       $result = $db->execute($sql);
+       $row = $db->fetchArray($result);
+       $num_feeds = (isset($row['CNT'])) ? $row['CNT'] : 0;
+       $num_bins = floor($num_feeds/$feeds_one_go) + 1;
+       $hour = date('H', $time);
+       $current_bin = $hour % $num_bins;
+
+
+       $limit = $db->limitOffset($limit, $feeds_one_go);
+       $sql = "SELECT * FROM MEDIA_SOURCE WHERE (TYPE='rss'
+          OR TYPE='html') $limit";
+       $result = $db->execute($sql);
+       $i = 0;
+       while($feeds[$i] = $this->db->fetchArray($result)) {
+         if($feeds[$i]['TYPE'] == 'html') {
+           list($feeds[$i]['CHANNEL_PATH'], $feeds[$i]['ITEM_PATH'],
           $feeds[$i]['TITLE_PATH'], $feeds[$i]['DESCRIPTION_PATH'],
           $feeds[$i]['LINK_PATH']) =
-          explode("###", html_entity_decode($feeds[$i]['AUX_INFO']));
+             explode("###", html_entity_decode($feeds[$i]['AUX_INFO']
+               ));
+         }
+         $i++;
       }
-      $i++;
-    }
-       unset($feeds[$i]); //last one will be null
+           unset($feeds[$i]); //last one will be null
+         }
        $feeds = FetchUrl::getPages($feeds, false, 0, NULL, "SOURCE_URL",
           CrawlConstants::PAGE, true, NULL, true);
        $feed_items = array();
@@ -648,4 +676,4 @@ class SourceModel extends Model
        return $meta_ids;
    }
 }
- ?>
\ No newline at end of file
+ ?>
--
1.9.4.msysgit.1
```