УДК 004.042

ВЕБ-ОРІЄНТОВАНА СИСТЕМА ДЛЯ УПРАВЛІННЯ МЕДІА-ФАЙЛАМИ ІНТЕРНЕТ-МАГАЗИНІВ

Качковський О.В.[1)], Крепич С.Я.[2)]
Тернопільський національний економічний університет
[1)]магістрант; [2)]к.т.н.. старший викладач

I. Вступ

Проблема управління медіа-файлами знайома усім власникам та адміністраторам веб-сайтів та онлайн-магазинів, однак особливо гостро її відчувають саме останні. Це пов'язано зі специфікою таких магазинів, адже практично усі сутності мають файли того чи іншого формату: продукти мають зв'язки із зображеннями в галереї; замовлення мають зв'язки із файлами для завантажування; користувачі магазину можуть налаштовувати аватари у своїх облікових записах , завантажувати файли зі своїх замовлень , обмінюватися файлам на форумах, у коментарях чи приватних повідомленнях.

Зрозуміло, що керувати такою кількістю файлів у "ручному" режимі не зручно та не безпечно, адже можна порушити зв'язок між сутністю та пов'язаними із нею файлами. Саме для підвищення продуктивності та безпечності цього процесу було створено вищезгадану систему управління медіа-файлами. З часом виникла потреба у підвищенні її продуктивності і було прийнято рішення щодо необхідності використання паралелізму у веб-орієнтованій системі.

II. Мета роботи

Метою роботи є вдосконалення наявної системи управління файлами онлайн-магазину на основі зміни алгоритму управління даними із використанням паралельних обчислень.

III. Постановка задачі

Дана система розроблена із використанням технології PHP. З моменту появи, PHP постачається з потоково-безпечною архітектурою, що реалізується шляхом використання декількох екземплярів цього інтерпретатора в окремих потоках у багатопотокових середовищах SAPI (Server API) [1]. Однак готового механізму для створення багатопотокових додатків «з коробки» на даний момент не існує. Для вирішення вказаної проблеми скористаємось бібліотекою *Pthreads* - об'єктно-орієнтованим API, який дозволяє використовувати створені користувачами потоки в PHP [2]. *Pthreads* включає в себе усі інструменти, необхідні для створення багатопоточних додатків, націлених на веб або консоль [3].

Нижче наведено офіційний зразок коду для ознайомлення з бібліотекою Pthreads:

```
<?php
class SomeThreadedClass extends Thread
{
 private $tID;
 public $data;
 public function __construct(int $tID)
{
 $this->tID = $tID;
 $this->data = $tID.":".date('H:i:s');
}
 public function run()
{
 echo $this->tID . " started.\n";
 sleep($this->tID);
 echo $this->tID ."ended.".date('H:i:s')."\n";
}
}
$threads = [];
for ($i = 1; $i < 5; $i++) {
    $threads[$i] = new SomeThreadedClass($i);
    $threads[$i]->start();
}
for ($i = 1; $i < 5; $i++) {
    $threads[$i]->join();
    echo $threads[$i]->data . "\n";
}
```

Основною відмінністю нової версії системи є зміна в алгоритмі опрацювання даних. Зміни полягають у створенні "ядра", яке буде викликатися у кількох "потоках" та створенні методу, що синхронізуватиме ці потоки.

Нижче наведено лістинг коду ядра:

```
function downloadImage($source, $target, $step
= null){
  preg_match('/(.+)\/([^\/]+)$/', $target,
$targetUrlPart);
  exec('mkdir -p ' . $targetUrlPart[1]);
  exec('curl -Lk "' . $source . '" > "' .
$target . '"');
  $img = @file_get_contents($target);

  if (!$img || preg_match('/(<html)/', $img))
{
    $arrContextOptions = array(
      'ssl' => array(
        'verify_peer'      => false,
        'verify_peer_name' => false,
      ),
    );
    file_put_contents($target,
@file_get_contents($source, false,
stream_context_create($arrContextOptions)));
    $img = @file_get_contents($target);
  }
  $fileSize = validateImage($img, $target);
  if (!$fileSize) {
```

```
$target)) {
    return $fileSize;
  } else {
    if (USING_PROXY) {
      $proxyArray = explode(', ',
PROXY_SERVER);
      $proxy = $proxyArray[rand(0,
count($proxyArray) - 1)];
      exec('curl -x ' . $proxy . ' --proxy-
user ' . PROXY_AUTH . ' -L ' . $source . ' > '
. $target);
      $img = @file_get_contents($target);
      if ($fileSize = validateImage($img,
$target)) {
        return $fileSize;
      }
    }
  }
  if ($step !== 10) {
    $step = $step ? ++$step : 1;
    sleep(2 * $step);
    return downloadImage($source, $target,
$step);
  }
  exec('rm -Rf ' . $target);
  return 0;
}
```

Даний метод оголошено в класі *ImageDownloader*, що наслідує визначений у *pthread* клас *Thread*. Він відповідає за найважливіші функції системи, що вдосконалюється, а саме завантаження та збереження файлів. Під час створення новий потік додається до загального масиву потоків та викликається за допомогою методу *start*. Використання додаткових потоків дозволяє не лише пришвидшити процес завантаження файлів, але і зменшити навантаження на систему.

IV. Програмна реалізація

Коротко ознайомимося з основними можливостями та користувацьким інтерфейсом системи. На стартовій сторінці системи користувачеві надається можливість обрати каталог для якого почнеться завантаження файлів. На рисунку 1 проілюстровано процес вибору каталогу. На рисунку 2 проілюстроване головне вікно системи після початку процесу завантаження файлів.

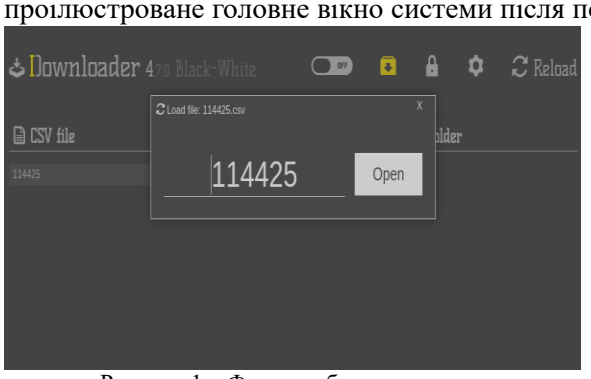
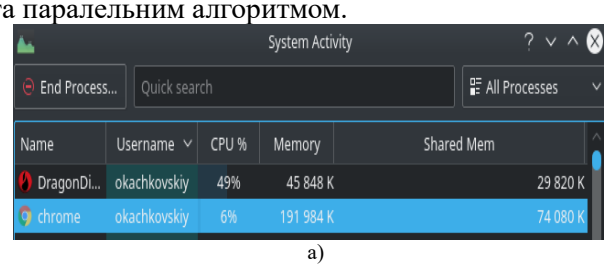


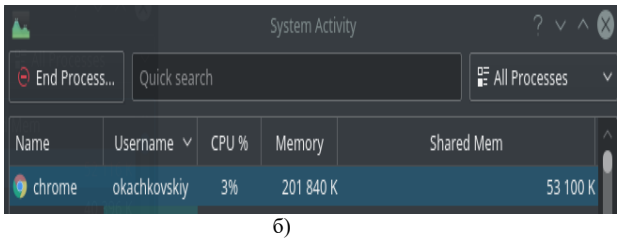
Рисунок 1 – Форма вибору каталогу для завантаження файлів

Рисунок 2 – Головна сторінка системи під час завантаження файлів

Для порівняння використання ресурсів ПК (див.рис.3) завантажемо 1000 файлів послідовним та паралельним алгоритмом.



а)



б)

Рисунок 2 - Навантаження на ПК за умови: а) послідовного завантаження файлів та б) паралельного завантаження файлів

Висновок

У роботі розглянуто спосіб оптимізації веб-орієнтованої системи для управління файлами інтернет-магазинів, створеної із використанням технології PHP за допомогою використання багатьох потоків. Із порівняння використання ресурсів ПК видно, що використання багатьох потоків не лише прискорює процес завантаження файлів, але й вдвічі зменшує навантаження на центральний процесор ПК.

Список використаних джерел

1. Сайт технології PHP. [Електронний ресурс]. Режим доступу - http://php.net
2. Довідник бібліотеки pthreads. [Електронний ресурс]. Режим доступу – http://php.net/manual/en/book.pthreads.php
3. Parallel Programming with Pthreads in PHP – the Fundamentals. [Електронний ресурс]. Режим доступу - https://www.sitepoint.com/parallel-programming-pthreads-php-fundamentals/