Porting LAMP Applications to Vibe.d

By Steven Schveighoffer

Daddy, I wish your web site ran on vibe.d

Me too, Pumpkin. Me too.
What is a LAMP?

- **L** - Linux
- **A** - Apache web server
- **M** - MySQL database (Can be any type of DB really)
- **P** - PHP language (Also can mean Perl, Python, Ruby)
A Bit of History

- 2009: Joined NRM, a company specializing in installing smart cooler controls. www.nrminc.com

- Charged with creating an internal job tracking system

- LAMP application called “Willow Job Tracker” already in use in one division.

- Had very little web development experience.

- D at version 2.032, no real Web frameworks available
CoolTrac was Born

- Customized Willow Job Tracker to fit the workflow needs of our company
- Lots of back-end improvements over original design to ease my suffering.
- Introduced jQuery and jQueryUI for more “live” UI.
DConf 2013

• Vladimir Panteleev’s presentation on D for Web Development was eye opening

• Began considering how I could use vibe.d instead of PHP

• Fateful conversation with Andrei
Why Vibe.d?

1. Performance
2. Statically typed
3. REST support
4. Metaprogramming
5. Because it’s D!
Porting Requirements

1. This is an application used every day, we must keep it running while doing development

2. Keep the same database model including authentication

3. Seamless integration between D and PHP sides *from a user perspective*

4. Fix UI issues by redoing all CSS/javascript and standardizing on Bootstrap
Porting Plan

- System config
  - Apache
  - PHP Session management
- Back-end
  - Data model
  - Authentication and messages
  - Linking between back-ends
- Front end
  - Navigation
  - Mobile support
Porting Plan

- **System config**
  - Apache
    - PHP Session management
  - Back-end
    - Data model
    - Authentication and messages
    - Linking between back-ends
- Front end
  - Navigation
  - Mobile support
Apache Config

- Apache handles all requests, vibe.d is handled via reverse proxy.

- Dedicate a subdirectory to vibe.d requests that isn’t already used, “/d/”.

- Use **ProxyPass** and **ProxyPassReverse** directives

  ```
  ProxyPass /d/ http://127.0.0.1:8080/
  ProxyPassReverse /d/ http://127.0.0.1:8080/
  <Location /d/>
      ProxyPassReverse /
  </Location>
  ```
Apache Config

• Want to avoid any reliance on Apache config in vibe.d itself

• Use mod_proxy_html to rewrite all links outside of headers

```xml
<Location /d/>
  SetOutputFilter proxy-html
  ProxyPassReverse /
  ProxyHTMLURLMap / /d/
</Location>
```

• Ensure you have the file `proxy_html.conf` under `mods_enabled` directory!
Porting Plan

- **System config**
  - Apache

- **PHP Session management**

- **Back-end**
  - Data model
  - Authentication and messages
  - Linking between back-ends

- **Front end**
  - Navigation
  - Mobile support
PHP Session Config

• By default PHP puts sessions into a directory.

• php-redis package usually available. On Ubuntu, it’s php-redis.

• Easy to config PHP to use Redis.

```
[Session]
; Handler used to store/retrieve data.
; http://php.net/session.save-handler
; session.save_handler = files
session.save_handler = redis
...
; session.save_path = "/var/lib/php5"
session.save_path = "tcp://127.0.0.1"
```

• Use `php_serialize` as serialization method

```
; Handler used to serialize data. php is the standard serializer of PHP.
; http://php.net/session.serialize-handler
session.serialize_handler = php_serialize
```
PHP Session Config

- Redis is by default insecure
- Offers simple password security, but no encryption or any other mechanisms for rejecting hackers.
- Use UNIX sockets when available - supports filesystem permission protections.
Porting Plan

• System config
  • Apache
  • PHP Session management

• Back-end
  • Data model
    • Authentication and messages
    • Linking between back-ends

• Front end
  • Navigation
  • Mobile support
Data Model

- MySQL is supported with mysql-native in vibe.d
- CoolTrac’s data system is hand-written SQL using string interpolation
- ORM is not easy because of the hand-tuned queries used to fetch data.
- Wanted something more usable than arrays of Variants (mysql-native’s interface)
Data Model

• Serialization library! Full utilization of D capabilities:

```d
struct Person
{
    int id;
    string username;
    @(DB.colName!"password") string passwordHash;
    @(DB.colName!"status", DB.colType!string, DB.colParser!setAIBool) bool active;
    DateTime lastupdate;
    DateTime createdon;
    string firstname;
    string lastname;
    ...
}
```

• Used via a nice range syntax that produces typed items from database rows:

```d
int foo(Connection conn)
{
    import std.stdio;
    foreach (p; conn.query("SELECT * FROM `person` WHERE `city` = ?", "Munich")
        .byItem!Person)
        writeln("%s %s is from Munich", p.firstname, p.lastname);
}
```
Data Model

- Query building tools:

```python
auto sql = "timesheet AS ts"
    .sqlJoin("person AS p ON (p.id = ts.person_id)"
    .select("ts.*")
    .where("ts.year = ?", nYear)
    .where("ts.week = ?", nWeek)
    .where("ts.person_id = ?", forPerson)
    .where("ts.approved_on IS NOT NULL", !includeUnapproved)
    .where("ts.latest = 1")
    .orderBy("ts.year", "ts.week", "p.lastname", "p.firstname")
return conn.query(sql);
```
Porting Plan

• System config
  • Apache
  • PHP Session management

• Back-end
  • Data model
  • Authentication and messages
    • Linking between back-ends

• Front end
  • Navigation
  • Mobile support
Authentication and Messages

- Single sign-on for both PHP and D
- Need to have each side be able to read the other’s cookies and session.
- Logging out is simple — destroy both sessions!
Process vibe.d Data in PHP

// see if there is any message from the vibe.d side of things
if(isset($_COOKIE['vibe_session_id']))
{
    $vibed_sessionid = $_COOKIE['vibe_session_id'];
    // use redis to get at the vibe session data
    $redis = new Redis();
    if($redis->connect("127.0.0.1"))
    {
        $redis->select(0);
        $extraSessionData = $redis->hGet($vibed_sessionid, "phpSessionData");
        if($extraSessionData !== FALSE)
        {
            // clear the session data
            $redis->hDel($vibed_sessionid, "phpSessionData");

            // process the session data.
            $sessionArr = json_decode($extraSessionData, true);
            if($sessionArr !== NULL)
                $_SESSION = array_merge($_SESSION, $sessionArr);
        }
    }
    $redis->close();
}
// read the user information. If user does not exist or is invalid, // close the session.
int userid = req.session.get!(int)("userid", -1);
if(userid == -1)
{
    // in this case, see if the php's session is logged in.
    try
    {
        if(auto phpsessid = req.cookies.get("PHPSESSID"))
        {
            auto redisConn = connectRedisDB(URL("redis://127.0.0.1/0"));
            scope(exit) redisConn.client.quit();
            // php session is stored "PHPREDIS_SESSION:<cookie>"
            auto phpSessionString = redisConn.get("PHPREDIS_SESSION:" ~ phpsessid);
            if(phpSessionString.length)
            {
                ...
            }
        }
    }
}
PHP Session Parser

- Yay, more serialization!

```plaintext
import nrm.phpser;
import std.algorithm: findSkip;
import std.range: retro;
static struct PHPSessionData
{
    string wjtpsession;
}
PHPSessionData sdata;
phpDeserialize/phpSessionString, sdata);
auto runtime = sdata.wjtpsession.retro;
enforce(runtime.findSkip("_")// timestamp
enforce(runtime.findSkip("_")// hash
auto uname = runtime.retro;
enforce(uname.findSkip("_")// junk

// get the username from the person database
auto u = conn.fetchPersonByUsername(uname);
req.session.set!(int)("userid", u.id);
userid = u.id;
```
Porting Plan

- System config
  - Apache
  - PHP Session management

- **Back-end**
  - Data model
  - Authentication and messages

- **Linking between back-ends**

- Front end
  - Navigation
  - Mobile support
More Apache Config

• We need an override for the proxy!

  ProxyPass  /d/PHPExt/ !
  Redirect  /d/PHPExt/ /cooltrac/www/
  ProxyPass  /d/ http://127.0.0.1:8080/
  ProxyPassReverse  /d/ http://127.0.0.1:8080/
Use Redirects Everywhere

• Links are sprinkled all over a web application.

• In D-land, we use placeholders to prevent having to do work when the target is ported.

• In PHP-land, we use redirects to avoid doing work once a page is ported.
Use Redirects Everywhere

- Example in CoolTrac: in PHP the URL for job 12345 is
  
  /cooltrac/www/index.php?display=JobDetails&id=12345

- Implement a placeholder route in vibe.d:

  /job/12345

```d
@path("/job/")
class JobWebImpl {
  @path(":jobId")
  void getJob(HTTPServerRequest req, HTTPServerResponse res, int _jobId) {
    import std.format;
    res.redirect(
      format("/PHPExt/index.php?display=JobDetails&id=%d", _jobId));
  }
}
Use Redirects Everywhere

// URL must be complete, not relative
$prot = ( "443" ==)$_SERVER["SERVER_PORT"] ) ? "https://" : "http://";
if(substr($this->url, 0, 1) === "/")
{
    $fullURL = $prot .$_SERVER["SERVER_NAME"] . $this->url;
}
else if(substr($this->url, 0, 4) === "http")
{
    $fullURL = $this->url;
}
else
{
    $self = $_SERVER["PHP_SELF"];
    $herepath = preg_replace("/[^\/]+$/", ",", $self);
    if (!preg_match("/^$/", $herepath))
    {
        $herepath .= "/";
    }
    $fullURL = $prot .$_SERVER["SERVER_NAME"] . $herepath . $this->url;
}
header("Location: $fullURL", true, $this->responseCode);
exit();
class JobDetails extends RedirectDisplay // use to extend TemplateDisplay
{
    function getName()
    {
        return "JobDetails";
    }

    function initialize()
    {
        $id = $this->context->request->get( "id" );
        $this->url = "/d/job/$id";
    }
}
Porting Plan

- System config
  - Apache
  - PHP Session management
- Back-end
  - Data model
  - Authentication and messages
  - Linking between back-ends
- Front end
  - Navigation
  - Mobile support
Navigation Design

• Should be similar in structure as the original to avoid confusion.

• Provide the top links the users are used to — you can always change it after the port is done.

• Perhaps take this time to improve with the D side of things.
Comparison of CoolTrac

- Before:

- After:
Porting Plan

- System config
  - Apache
  - PHP Session management
- Back-end
  - Data model
  - Authentication and messages
  - Linking between back-ends
- Front end
  - Navigation
- Mobile support
Congratulations!

- Now you have a vibe.d app to impress all your friends!
- Need a good acronym...

LAVDRMP
Live Demo!
Diamond MVC

- Runs on top of Vibe.d
- Provides alternative system to vibe.d’s diet templates
- Adds more framework for controllers/ORM
Symfony

- A more sophisticated, modernized, and supported framework than what CoolTrac is based on.

- Almost everything in this presentation still applies
Symfony - YAML

- Lots of configuration is stored in yaml files
- D has yaml support!
Symfony - ORM

- Symfony uses an ORM called doctrine.
- Schema is unfortunately generated using PHP attributes.
- D has the potential to parse these, or using PHP, you could generate an alternative format.
- Use a supported D ORM instead: [http://code.dlang.org/search?q=ORM](http://code.dlang.org/search?q=ORM)
If you use YAML to do routing, updating links are easy.

```yaml
#load_job:
#    path: /job/{jobid}
#    controller: App\Controller\JobController::jobDetails
load_job:
    path: /job/{jobid}
    controller: App\Controller\Redirector::redirectJob
```
<?php

// src/Controller/Redirector.php
namespace App\Controller;

use Symfony\Bundle\FrameworkBundle\Controller\Controller;
use Symfony\Component\HttpFoundation\RedirectResponse;
use Symfony\Component\HttpFoundation\Request;

class Redirector extends Controller{

    public function redirectVibed(Request $request, $subroute)
    {
        $urldata = parse_url($request->getUri());
        $portopt = isset($urldata['port']) ? ':' . $urldata['port'] : '';
        return $this->redirect($urldata['scheme'] . '://' . $urldata['host'] . $portopt . '/d/' . $subroute);
    }

    public function redirectJob(Request $request, $jobid)
    {
        return $this->redirectVibed($request, "job/$jobid");
    }
}
Ditching Symfony — Consider the Drawbacks

- Command-line code generation
- Development bar
- Exception/Error display
- Built-in Authentication support
Some Notes

• Mysql-native has gotten a whole lot better, but I still find the serialization part a bit wanting. The row struct is also quite limited.

• DMD compilation is fast. Vibe.d compilation is not. Would love to see some improvement here.

• Having to recompile/rebuild the entire project to change some classes on a template.
Ever since I switched to vibe.d, developing my website has been so much more fun!

I hope someday I can make vibe.d websites just like you.

Thanks, Vibe.d!

Ask your Doctor if LAVDRMP is right for you
• Vladimir Panteleev’s Dconf 2013 presentation: http://dconf.org/2013/talks/panteleev.html

• Using Redis as a PHP session handler: https://www.digitalocean.com/community/tutorials/how-to-set-up-a-redis-server-as-a-session-handler-for-php-on-ubuntu-14-04

• Apache config: mod_proxy: https://httpd.apache.org/docs/2.4/mod/mod_proxy.html, mod_proxy_html: https://httpd.apache.org/docs/2.4/mod/mod_proxy_html.html.html (good luck understanding that!)

• Description of php_serialize format: http://www.phpinternalsbook.com/classes_objects/serialization.html

• Symfony: https://symfony.com/