Performance White Paper

This study benchmarks vanilla Magento hosted on the Swoole environment against the standard FastCGI infrastructure. The objective is to measure the performance difference between the equivalent environments on identical hardware.

Subject:

1. **Reference:** Magento 2.3.1 Open Source, small merchant profile, PHP-FPM
2. **Candidate:** Magento 2.3.1 Open Source, small merchant profile, Swoole conservative strategy

Scope: Frontend user activity scenarios

Criteria: Magento response time, Concurrency

Metrics: Latency

Methodology

Performance testing is conducted by emulating storefront user activity scenarios via the Magento Performance Toolkit. Magento application deployed from scratch has been populated with the fixture data of the small merchant profile.

User activity scenarios implemented as JMeter scripts of the Magento Performance Toolkit are used w/o modifications. The scenarios are repeated for 12000 loops executed with concurrency 10, 30, 60, 100 simultaneous users:

- 10 users – 1200 loops
- 30 users – 400 loops
- 60 users – 200 loops
- 100 users – 120 loops

Latency recorded by JMeter includes response generation time by Magento and data transfer time over the network. The measurements are taken from within a local network to approximate the Latency to the server response time.

Full Page Cache (FPC) has been disabled to focus testing on the server-side performance of the Magento application. Remaining Magento caches are enabled according to the standard deployment practices.

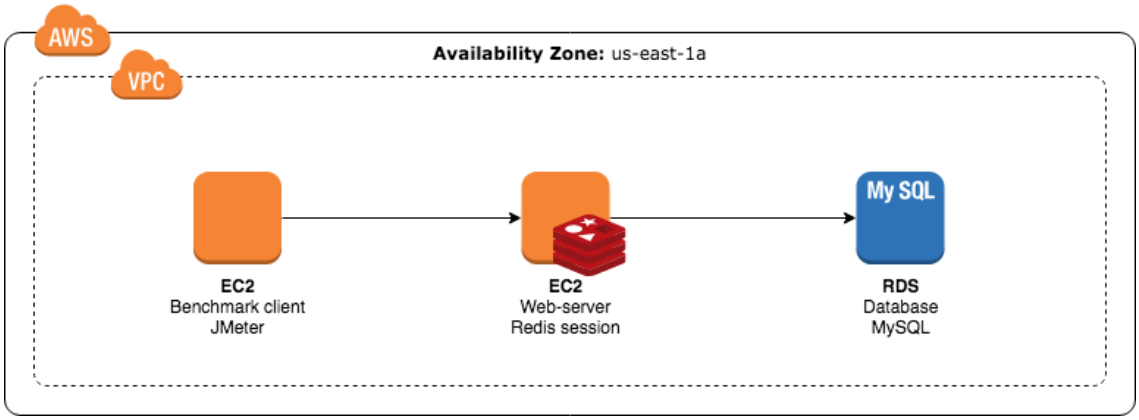
Infrastructure

Virtualized AWS Cloud infrastructure composed of EC2 and RDS instances is utilized for hosting and testing.

Single web-node of reasonable capacity is used to minimize influence of networking factors on testing results.

Redis server is used as a session storage instead of the filesystem – a setup typical to multiple web-node infrastructure. The reason to utilize Redis session storage with a single web-node is ability to control the session locking behavior. Session locking has been disabled to avoid its performance overhead as a common performance tuning practice.

Varnish caching server is absent in accordance with the disabled FPC and testing objectives.



Hardware

Instance	Hardware	vCPU (clock speed)	RAM, GiB	Network, Gbps	Storage, IOPS
Database	RDS db.m5.2xlarge	8 (2.5 GHz)	32	up to 10	2,000
Web-server	EC2 c5.4xlarge	16 (3.0–3.5 GHz)	32	up to 10	20,000
Benchmark client	EC2 c5.4xlarge	16 (3.0–3.5 GHz)	32	up to 10	20,000

Software

Instance	Common Software	Reference Software	Candidate Software
Database	MySQL 5.7.25 Community		
Web-server	Amazon Linux 2 Nginx 1.12.2 PHP 7.1.27 Zend OPcache 7.1.27 APCu 5.1.17 Redis 3.2.12 Magento 2.3.1 Open Source	PHP-FPM 7.1.27	Swoole 4.3.1 Upscale_Swoole 5.0.1
Benchmark client	Amazon Linux 2 Java JDK 1.8.0 JMeter 3.1 JMeter plugin JSON 2.5 Magento Performance Toolkit 2.3.1		

Benchmark

Results

Swoole conservative strategy outperforms PHP-FPM in all scenarios:

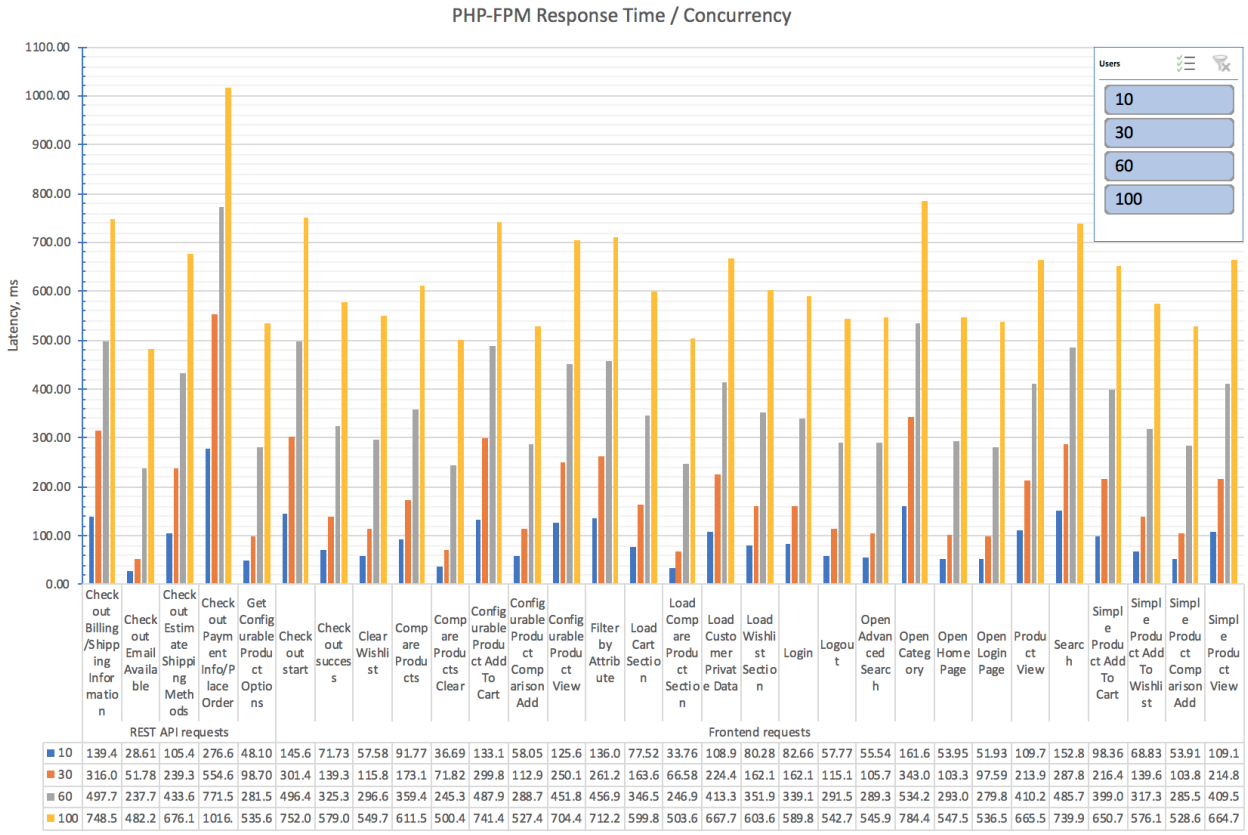
- Swoole is **~35% faster** than PHP-FPM at any concurrency 10, 30, 60, 100 users
- Swoole is **~45% faster** than PHP-FPM on REST API requests at concurrency 10, 30 users
- Swoole is **~65% faster** than PHP-FPM on REST API requests at concurrency 60 users
- Swoole is **~75% faster** than PHP-FPM on REST API requests at concurrency 100 users

Sampling

Execute JMeter scenarios in non-GUI mode on the Benchmark client instance:

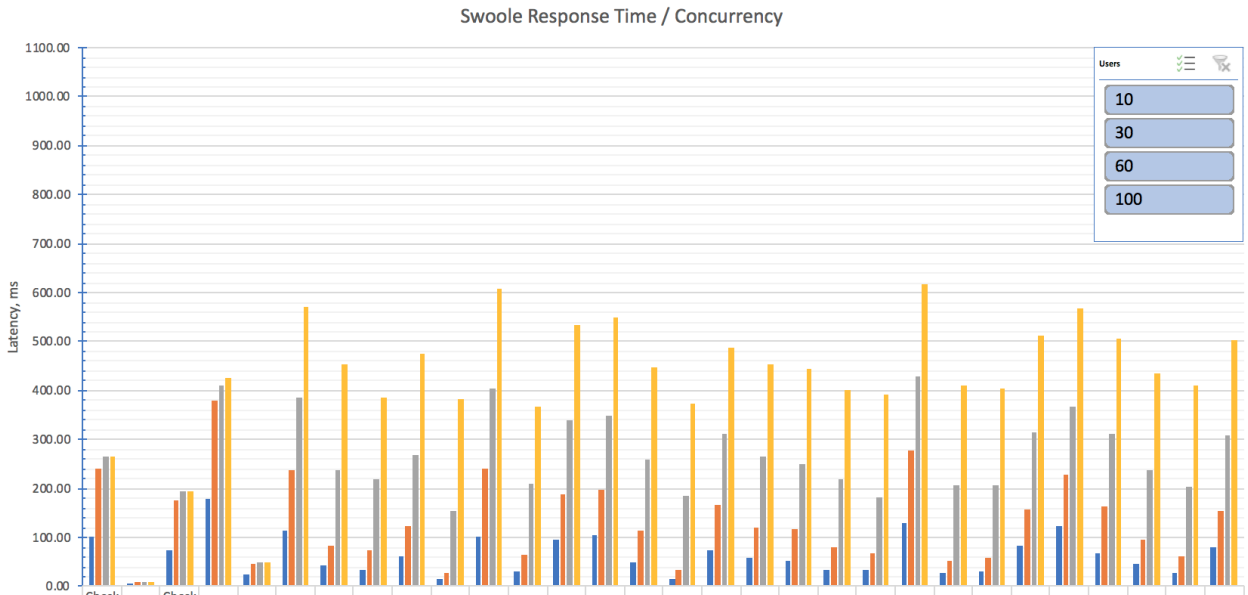
```
jmeter -n \
    -t setup/performance-toolkit/benchmark.jmx \
    -j magento231_10users_1200loops.log \
    -l magento231_10users_1200loops.csv \
    -Jhost=http://magento2-demo1.upscalesoftware.com \
    -Jbase_path=/ \
    -Jadmin_path=admin \
    -Jadmin_user=admin \
    -Jadmin_password=123123q \
    -Jfiles_folder=setup/performance-toolkit/files/ \
    -Jloops=1200 \
    -JfrontendPoolUsers=10 \
    -JadminPoolUsers=0 \
    -JapiPoolUsers=0 \
    -JcsrPoolUsers=0 \
    -JdeadLocksPoolUsers=0 \
    -JothersPoolUsers=0
```

Reference Samples



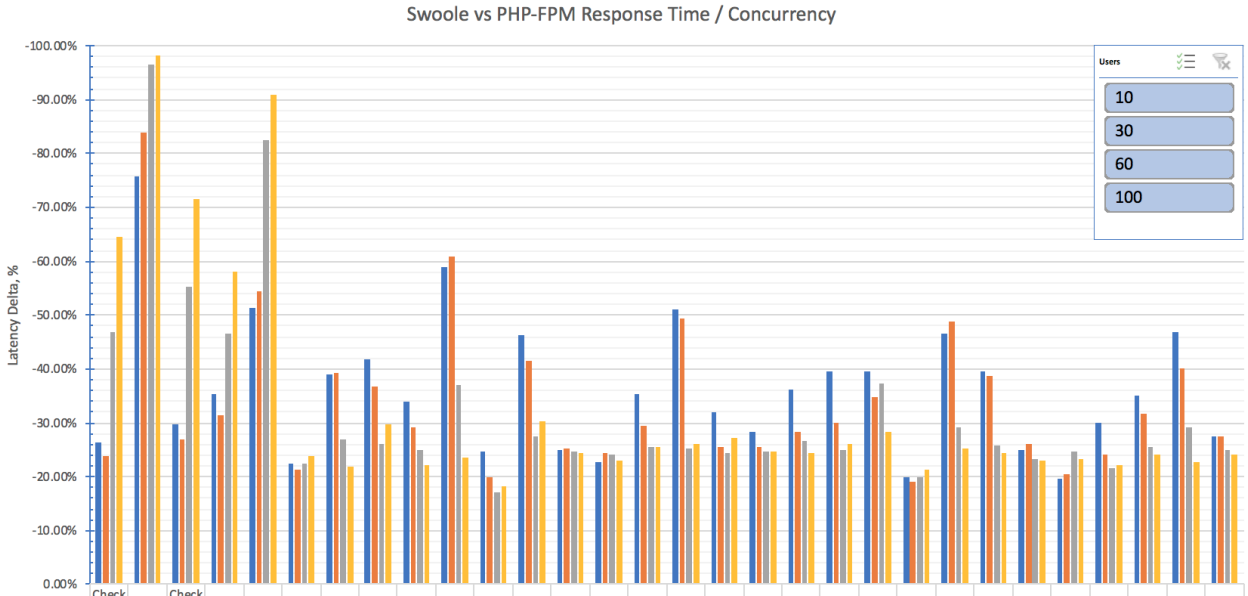
	Checkout Billing/Shipping Information	Checkout Email Available	Checkout Estimate Shipping Methods	Checkout Payment Info/Place Order	Get Configurable Product Options	Checkout start	Checkout success	Clear Wishlist	Compare Products	Compare Products Clear	Configurable Product Add To Cart	Configurable Product Comparison Add	Configurable Product View	Filter by Attribute	Load Cart Section	Load Compare Product Section	Load Customer Private Data	Load Wishlist Section	Login	Logout	Open Advanced Search	Open Category	Open Home Page	Open Login Page	Product View	Search	Simple Product Add To Cart	Simple Product Add To Wishlist	Simple Product Comparison Add	Simple Product View
10	139.4	28.61	105.4	276.6	48.10	145.6	71.73	57.58	91.77	36.69	133.1	58.05	125.6	136.0	77.52	33.76	108.9	80.28	82.66	57.77	55.54	161.6	53.95	51.93	109.7	152.8	98.36	68.83	53.91	109.1
30	316.0	51.78	239.3	554.6	98.70	301.4	139.3	115.8	173.1	71.82	299.8	112.9	250.1	261.2	163.6	66.58	224.4	162.1	162.1	115.1	105.7	343.0	103.3	97.59	213.9	287.8	216.4	139.6	103.8	214.8
60	497.7	237.7	433.6	771.5	281.5	496.4	325.3	296.6	359.4	245.3	487.9	288.7	451.8	456.9	346.5	246.9	413.3	351.9	339.1	291.5	289.3	534.2	293.0	279.8	410.2	485.7	399.0	317.3	285.5	409.5
100	748.5	482.2	676.1	1016.	535.6	752.0	579.0	549.7	611.5	500.4	741.4	527.4	704.4	712.2	599.8	503.6	667.7	603.6	589.8	542.7	545.9	784.4	547.5	536.5	665.5	739.9	650.7	576.1	528.6	664.7

Candidate Samples



	Checkout Billing/Shipping Information	Checkout Email Available	Checkout Estimate Shipping Methods	Checkout Payment Info/Place Order	Get Configurable Product Options	Checkout start	Checkout success	Clear Wishlist	Compare Products	Compare Products Clear	Configurable Product Add To Cart	Configurable Product Comparison Add	Configurable Product View	Filter by Attribute	Load Cart Section	Load Compare Product Section	Load Customer Private Data	Load Wishlist Section	Login	Logout	Open Advanced Search	Open Category	Open Home Page	Open Login Page	Product View	Search	Simple Product Add To Cart	Simple Product Add To Wishlist	Simple Product Comparison Add	Simple Product View
	REST API requests					Frontend requests																							
10	102.5	6.93	73.95	178.6	23.37	113.0	43.85	33.48	60.68	15.12	100.3	31.12	94.19	105.2	50.15	16.49	74.23	57.56	52.79	34.99	33.58	129.4	28.90	31.37	82.42	122.9	68.96	44.75	28.63	79.21
30	241.0	8.39	174.6	380.5	44.91	237.0	84.51	73.22	122.6	28.05	240.3	66.12	186.8	197.6	115.3	33.75	167.1	120.5	115.9	80.47	69.11	277.1	52.98	59.83	157.8	229.2	164.4	95.44	62.26	155.5
60	264.0	8.18	194.4	411.8	48.96	384.6	237.7	218.8	269.2	154.5	404.3	209.2	339.9	347.3	258.4	184.8	312.1	265.4	248.7	219.0	181.7	427.7	207.6	207.9	315.0	366.3	312.4	236.6	202.1	307.4
100	265.9	8.06	193.0	426.1	49.23	571.7	452.5	385.5	475.9	382.3	606.9	368.2	533.0	549.2	447.3	372.2	486.9	454.1	445.8	401.4	390.6	616.4	409.2	405.4	513.0	568.0	506.4	436.5	409.1	503.7

Comparison



	Checkout Billing/Shipping Information	Checkout Email Available	Checkout Estimate Shipping Methods	Checkout Payment Info/Place Order	Get Configurable Product Options	Checkout start	Checkout success	Clear Wishlist	Compare Products	Compare Products Clear	Configurable Product Add To Cart	Configurable Product Comparison Add	Configurable Product View	Filter by Attribute	Load Cart Section	Load Compare Product Section	Load Customer Private Data	Load Wishlist Section	Login	Logout	Open Advanced Search	Open Category	Open Home Page	Open Login Page	Product View	Search	Simple Product Add To Cart	Simple Product Add To Wishlist	Simple Product Comparison Add	Simple Product View
	REST API requests					Frontend requests																							
10	-26.4	-75.7	-29.8	-35.4	-51.4	-22.4	-38.8	-41.8	-33.8	-58.8	-24.5	-46.3	-25.0	-22.6	-35.3	-51.1	-31.8	-28.3	-36.1	-39.4	-39.5	-19.9	-46.4	-39.5	-24.9	-19.5	-29.8	-34.9	-46.8	-27.4
30	-23.7	-83.8	-27.0	-31.3	-54.5	-21.3	-39.3	-36.7	-29.1	-60.9	-19.8	-41.4	-25.3	-24.3	-29.4	-49.3	-25.5	-25.6	-28.4	-30.1	-34.6	-19.2	-48.7	-38.7	-26.1	-20.3	-24.0	-31.6	-40.0	-27.6
60	-46.9	-96.5	-55.1	-46.6	-82.6	-22.5	-26.9	-26.2	-25.1	-36.9	-17.1	-27.5	-24.7	-24.0	-25.4	-25.1	-24.4	-24.5	-26.6	-24.8	-37.1	-19.9	-29.1	-25.7	-23.2	-24.5	-21.7	-25.4	-29.2	-24.9
100	-64.4	-98.3	-71.4	-58.0	-90.8	-23.9	-21.8	-29.8	-22.1	-23.6	-18.1	-30.1	-24.3	-22.8	-25.4	-26.1	-27.0	-24.7	-24.4	-26.0	-28.4	-21.4	-25.2	-24.4	-22.9	-23.2	-22.1	-24.2	-22.6	-24.2

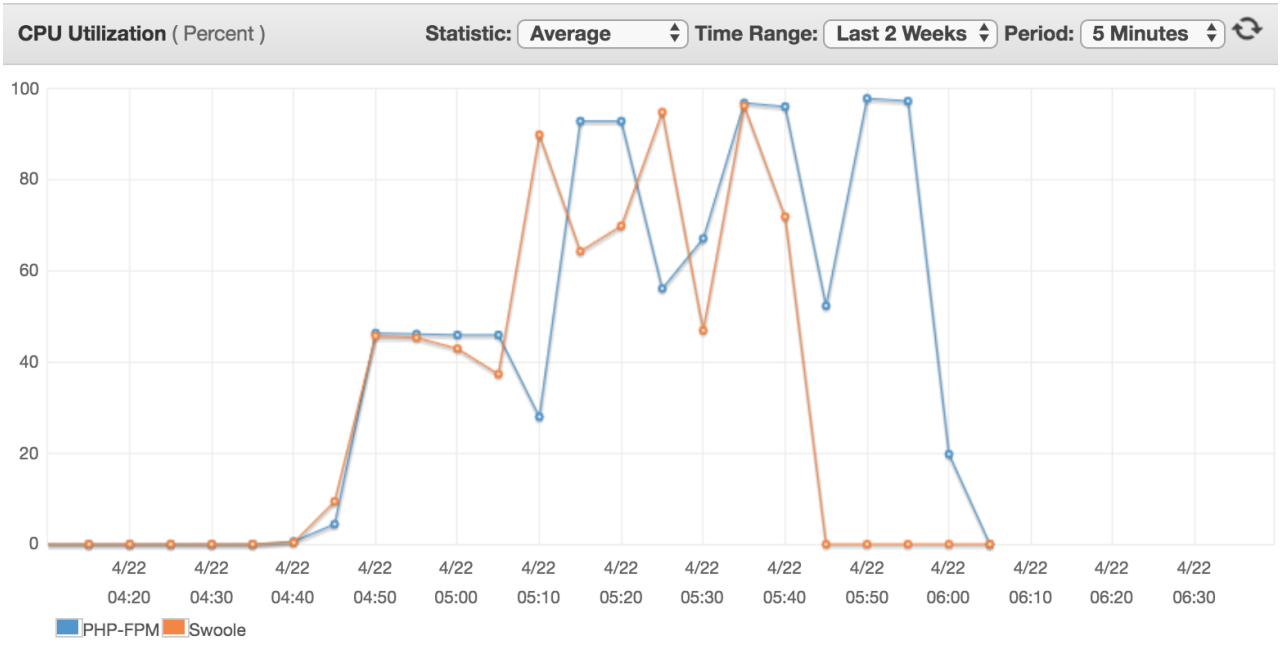
Average of Latency Delta	Column Labels			
Row Labels	10	30	60	100
REST API requests	**-43.80%**	**-44.09%**	**-65.58%**	**-76.63%**
Checkout Billing/Shipping Information	-26.47%	-23.75%	-46.94%	-64.48%
Checkout Email Available	-75.79%	-83.80%	-96.56%	-98.33%
Checkout Estimate Shipping Methods	-29.87%	-27.04%	-55.16%	-71.45%
Checkout Payment Info/Place Order	-35.44%	-31.38%	-46.63%	-58.09%
Get Configurable Product Options	-51.42%	-54.50%	-82.61%	-90.81%
Frontend requests	**-34.64%**	**-31.93%**	**-25.73%**	**-24.39%**
Checkout start	-22.40%	-21.34%	-22.51%	-23.97%
Checkout success	-38.87%	-39.37%	-26.92%	-21.85%
Clear Wishlist	-41.86%	-36.77%	-26.23%	-29.87%
Compare Products	-33.88%	-29.17%	-25.11%	-22.18%
Compare Products Clear	-58.80%	-60.94%	-36.99%	-23.60%
Configurable Product Add To Cart	-24.59%	-19.85%	-17.13%	-18.13%
Configurable Product Comparison Add	-46.39%	-41.46%	-27.52%	-30.18%
Configurable Product View	-25.06%	-25.30%	-24.77%	-24.33%
Filter by Attribute	-22.63%	-24.36%	-24.00%	-22.88%
Load Cart Section	-35.31%	-29.49%	-25.41%	-25.42%
Load Compare Product Section	-51.15%	-49.30%	-25.14%	-26.10%
Load Customer Private Data	-31.87%	-25.54%	-24.49%	-27.08%
Load Wishlist Section	-28.30%	-25.64%	-24.57%	-24.77%
Login	-36.14%	-28.46%	-26.66%	-24.42%
Logout	-39.44%	-30.11%	-24.85%	-26.02%
Open Advanced Search	-39.53%	-34.66%	-37.18%	-28.44%
Open Category	-19.96%	-19.21%	-19.94%	-21.42%
Open Home Page	-46.44%	-48.73%	-29.16%	-25.26%
Open Login Page	-39.59%	-38.70%	-25.71%	-24.42%
Product View	-24.90%	-26.19%	-23.20%	-22.92%
Search	-19.58%	-20.35%	-24.59%	-23.23%
Simple Product Add To Cart	-29.89%	-24.02%	-21.70%	-22.18%
Simple Product Add To Wishlist	-34.99%	-31.63%	-25.43%	-24.24%
Simple Product Comparison Add	-46.88%	-40.03%	-29.22%	-22.60%
Simple Product View	-27.41%	-27.63%	-24.93%	-24.23%
Grand Total	**-36.16%**	**-33.96%**	**-32.38%**	**-33.10%**

Observations:

- PHP-FPM response time increases proportionally to concurrency
- Swoole response time increases proportionally to concurrency, except for REST API requests
- Swoole response time of Frontend requests increases faster than PHP-FPM at concurrency 60, 100 users
- Swoole response time of REST API requests is constant at concurrency 30, 60, 100 users

Utilization

Web-server Resources



Concurrency	PHP-FPM CPU Utilization	Swoole CPU Utilization
10 users	~46%	~45%
30 users	~93%	~90%
60 users	~97%	~95%
100 users	~98%	~96%

Observations:

- 10 users – CPU capacity underutilized
- 30 users – CPU capacity saturated; min capacity buffer
- 60, 100 users – CPU capacity exceeded; excess requests queued

Configuration

The infrastructure has been optimized according to the following guides:

- AWS - Operating System Optimizations
- AWS - How Netflix Tunes EC2 Instances for Performance
- Nginx - Tuning NGINX for Performance
- Redis - 5 Tips for Running Redis over AWS
- Magento 2.3 - Software recommendations

Web-server Setup

OS Tuning

Amazon Linux 2 operating system has been optimized according to the hosting best practices.

Highlights:

- TSC clock source enforced
- CPU idle & wake up disabled
- RAM swapping to disk disabled
- Network timeouts increased
- Network buffers increased
- Connection backlog increased

Configuration:

- Boot loader config `/etc/default/grub` :

```
xen_nopvspin=1 clocksource=tsc tsc=reliable intel_idle.max_cstate=1
```

```
sudo grub2-mkconfig -o /boot/grub2/grub.cfg
```

- Memory config `/etc/sysctl.d/20-memory.conf`

```
# https://redislabs.com/blog/5-tips-for-running-redis-over-aws/
vm.swappiness = 0
vm.overcommit_memory = 1
# https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/enhanced-networking-os.html
vm.min_free_kbytes = 1048576
# http://www.brendangregg.com/Slides/AWSreInvent2017_performance_tuning_EC2.pdf
kernel.numa_balancing = 0
```

- Networking config `/etc/sysctl.d/20-networking.conf`

```
# http://www.brendangregg.com/Slides/AWSreInvent2017_performance_tuning_EC2.pdf
net.core.somaxconn = 1024
net.core.netdev_max_backlog = 5000
net.core.rmem_max = 16777216
net.core.wmem_max = 16777216
net.ipv4.tcp_wmem = 4096 12582912 16777216
net.ipv4.tcp_rmem = 4096 12582912 16777216
net.ipv4.tcp_max_syn_backlog = 8096
net.ipv4.tcp_slow_start_after_idle = 0
net.ipv4.tcp_tw_reuse = 1
#net.ipv4.tcp_tw_recycle = 1
#net.ipv4.tcp_abort_on_overflow = 1
```

- Filesystem config `/etc/sysctl.d/20-filesystem.conf`

```
# http://www.brendangregg.com/Slides/AWSreInvent2017_performance_tuning_EC2.pdf
vm.dirty_ratio = 80
vm.dirty_background_ratio = 5
vm.dirty_expire_centisecs = 12000
```

PHP Setup

Highlights:

- Zend OPcache extension installed
- OPcache memory and file limits increased
- OPcache file change detection disabled
- APCu shared memory extension installed (for autoload class map)

Configuration:

- OPcache config `/etc/opt/remi/php71/php.d/10-opcache.ini`

```
opcache.enable = 1
opcache.enable_cli = 1
opcache.memory_consumption = 512
opcache.interned_strings_buffer = 16
opcache.max_accelerated_files = 100000
opcache.validate_timestamps = 0
opcache.consistency_checks = 0
opcache.file_cache = /var/opt/remi/php71/lib/php/opcache
```

- APCu config `/etc/opt/remi/php71/php.d/40-apcu.ini`

```
apc.enabled = 1
```

Redis Setup

Highlights:

- Unix socket instead of TCP port
- Persistence to disk disabled
- Least recently used (LRU) records extrusion

Configuration:

- Redis config `/etc/redis.conf`

```
port 0
unixsocket /var/run/redis.sock
unixsocketperm 777
#save 900 1
#save 300 10
#save 60 10000
maxmemory 1G
maxmemory-policy allkeys-lru
maxmemory-samples 3
```

Nginx Setup

Highlights:

- Worker process count = CPU count
- Multiple connections per worker process
- Network I/O via epoll mechanism
- Access logging disabled

Configuration:

- Nginx config `/etc/nginx/nginx.conf`

```
user nginx;
worker_processes auto;
error_log /var/log/nginx/error.log crit;
pid /run/nginx.pid;

include /usr/share/nginx/modules/*.conf;

events {
    worker_connections  1024;
    multi_accept        on;
    use                 epoll;
}

http {
    access_log          off;

    sendfile            on;
    tcp_nopush          on;
    tcp_nodelay         on;
    types_hash_max_size 2048;

    include             /etc/nginx/mime.types;
    default_type        application/octet-stream;

    include /etc/nginx/conf.d/*.conf;
}
```

Magento Setup

Highlights:

- Magento production mode
- Magento DI compiled & code generated
- Magento static files deployed
- Dev dependencies not installed
- Autoload class map in APCu shared memory
- Filesystem cache storage
- Redis session storage
- Session locking disabled
- Full Page Cache (FPC) disabled
- Small merchant profile fixture data

Deployment steps:

1. Install dependencies excluding dev ones:

```
composer install --no-dev
```

2. Switch to the production mode (compile DI & deploy static):

```
php bin/magento deploy:mode:set production
```

3. Generate optimized autoload class map:

```
composer dump-autoload --no-dev --optimize --apcu
```

4. Generate small merchant profile fixture data:

```
php bin/magento setup:performance:generate-fixtures -s setup/performance-toolkit/profiles/ce/small.x
php bin/magento indexer:reindex
```

5. Disable Full Page Cache (FPC):

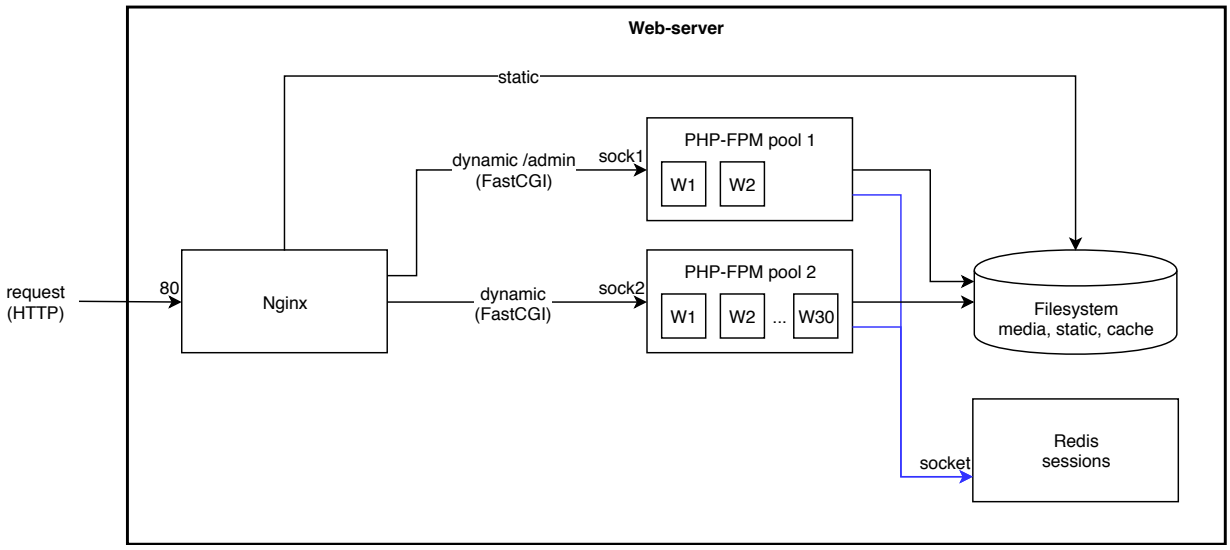
```
php bin/magento cache:disable full_page
php bin/magento cache:flush
```

Configuration:

- Magento deployment config `/var/www/magento2/app/etc/env.php`:

```php
<?php
return [
    // ...
    'MAGE_MODE' => 'production',
    'session' => [
        'save' => 'redis',
        'redis' => [
            'host' => '/var/run/redis.sock',
            'port' => '0',
            'password' => '',
            'timeout' => '2.5',
            'persistent_identifier' => '',
            'database' => '0',
            'compression_threshold' => '2048',
            'compression_library' => 'gzip',
            'log_level' => '3',
            'max_concurrency' => '6',
            'break_after_frontend' => '5',
            'break_after_adminhtml' => '30',
            'first_lifetime' => '600',
            'bot_first_lifetime' => '60',
            'bot_lifetime' => '7200',
            'disable_locking' => '1',
            'min_lifetime' => '60',
            'max_lifetime' => '2592000',
        ]
    ],
    'cache_types' => [
        'config' => 1,
        'layout' => 1,
        'block_html' => 1,
        'collections' => 1,
        'reflection' => 1,
        'db_ddl' => 1,
        'compiled_config' => 1,
        'eav' => 1,
        'customer_notification' => 1,
        'config_integration' => 1,
        'config_integration_api' => 1,
        'full_page' => 0,
        'config_webservice' => 1,
        'translate' => 1
    ],
];
```

Web-server FastCGI Setup



PHP-FPM Setup

Highlights:

- Unix socket instead of TCP port
- Network I/O via epoll mechanism
- Fixed number of worker processes
- Separate PHP-FPM process pools for admin and front/API
- Different timeouts for admin and front/API
- Different memory limit for admin and front/API

Configuration:

- PHP-FPM config `/etc/opt/remi/php71/php-fpm.conf`

```
log_level = warning
events.mechanism = epoll
```

- Front/API process pool config `/etc/opt/remi/php71/php-fpm.d/www.conf`

```
[www]
user = nginx
group = nginx

listen = /var/run/php-fpm.sock
listen.backlog = 1024
listen.owner = nginx
listen.group = nginx
listen.mode = 0660
listen.allowed_clients = 127.0.0.1

pm = static
pm.max_children = 30
pm.max_requests = 500

request_terminate_timeout = 60

php_admin_value[memory_limit] = 256M
```

- Admin process pool config `/etc/opt/remi/php71/php-fpm.d/admin.conf`

```
[admin]
user = nginx
```

```
group = nginx

listen = /var/run/php-fpm-admin.sock
listen.backlog = 1024
listen.owner = nginx
listen.group = nginx
listen.mode = 0660
listen.allowed_clients = 127.0.0.1

pm = static
pm.max_children = 2
pm.max_requests = 500

request_terminate_timeout = 7200

php_admin_value[memory_limit] = 1G
```

Nginx PHP-FPM Setup

Highlights:

- Static files served by Nginx
- Dynamic requests proxied to PHP-FPM process pools by URL

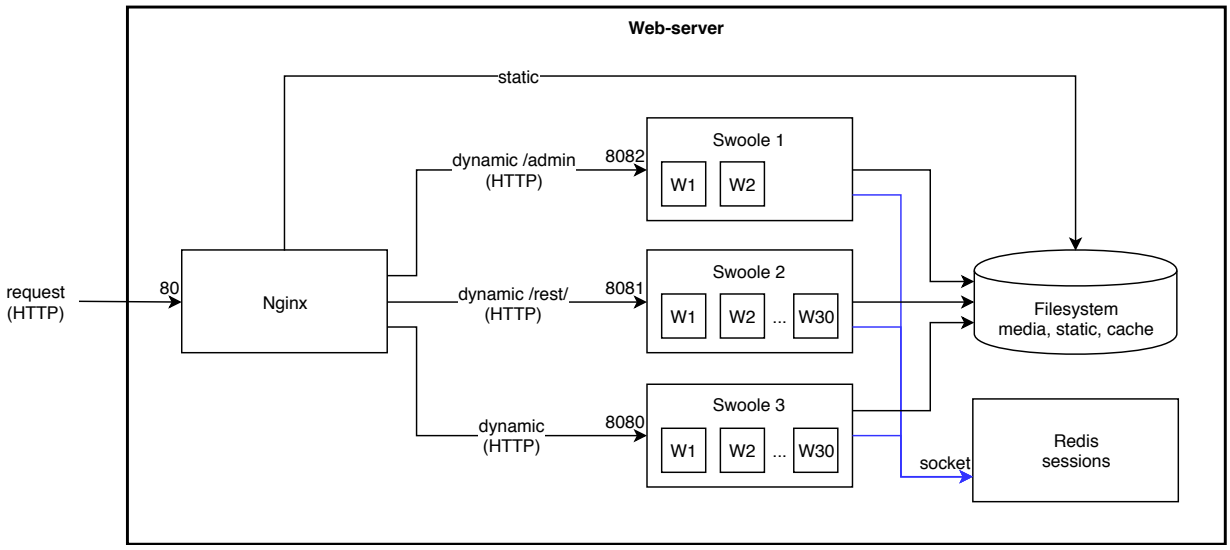
Configuration:

- Magento host config `/etc/nginx/conf.d/magento2.conf`

```
upstream magento_www {
    server unix:/var/run/php-fpm.sock;
}

upstream magento_admin {
    server unix:/var/run/php-fpm-admin.sock;
}

map $request_uri $fastcgi_backend {
    default                 magento_www;
    ~^(/index\.php)?/admin  magento_admin;
}

server {
    listen 80;
    server_name magento2-demo1.upscalesoftware.com;
    set $MAGE_ROOT /var/www/magento2;
    include /var/www/magento2/nginx.conf;
}
```

Config file `/var/www/magento2/nginx.conf` is a copy of `/var/www/magento2/nginx.conf.sample` shipped with Magento where all references `fastcgi_backend` have been replaced with the variable `$fastcgi_backend`.

Web-server Swoole Setup



Swoole Setup

Highlights:

- TCP port (Unix socket unsupported)
- Network I/O via epoll mechanism
- Swoole conservative strategy
- Separate Swoole instances for admin, REST API, and frontend
- Different timeouts for admin and front/API
- Different memory limit for admin and front/API

Configuration:

- Swoole admin instance config:
  - Swoole admin config `/etc/swoole-magento2-admin.ini`

```
worker_num = 2
max_conn = 1024
backlog = 4096

user = nginx
group = nginx

daemonize = 1
```

  - Swoole admin service `/etc/systemd/system/swoole-magento2-admin.service`

```
[Service]
Type=forking
ExecStart=/usr/bin/env php \
    -d memory_limit=1G \
    /var/www/magento2/vendor/bin/server.php \
    --name 'php-swoole [magento2 admin]' \
    -p 8082 \
    -x conservative \
    -a adminhtml \
    -f /etc/swoole-magento2-admin.ini
```

- Swoole REST API instance config:
  - Swoole API config `/etc/swoole-magento2-api.ini`

```
worker_num = 30
max_conn = 1024
backlog = 4096

user = nginx
group = nginx

daemonize = 1
```

  - Swoole API service `/etc/systemd/system/swoole-magento2-api.service`

```
[Service]
Type=forking
ExecStart=/usr/bin/env php \
    -d memory_limit=256M \
    /var/www/magento2/vendor/bin/server.php \
    --name 'php-swoole [magento2 api]' \
    -p 8081 \
    -x conservative \
    -a webapi_rest \
    -f /etc/swoole-magento2-api.ini
```

- Swoole frontend instance config:
  - Swoole frontend config `/etc/swoole-magento2-www.ini`

```
worker_num = 30
max_conn = 1024
backlog = 4096

user = nginx
group = nginx

daemonize = 1
```

  - Swoole frontend service `/etc/systemd/system/swoole-magento2-api.service`

```
[Service]
Type=forking
ExecStart=/usr/bin/env php \
    -d memory_limit=256M \
    /var/www/magento2/vendor/bin/server.php \
    --name 'php-swoole [magento2 www]' \
    -p 8080 \
    -x conservative \
    -a frontend \
    -f /etc/swoole-magento2-www.ini
```

Nginx Swoole Setup

Highlights:

- Static files served by Nginx
- Dynamic requests proxied to Swoole instances by URL

Configuration:

- Magento host config `/etc/nginx/conf.d/magento2-swoole.conf`

```
upstream magento_www {
    server 127.0.0.1:8080;
}

upstream magento_api {
    server 127.0.0.1:8081;
}
```

```
upstream magento_admin {
    server 127.0.0.1:8082;
}

map $request_uri $swoole_backend {
    default                 http://magento_www;
    ~^(/index\.php)?/rest/  http://magento_api;
    ~^(/index\.php)?/admin  http://magento_admin;
}

server {
    listen 80;
    server_name magento2-demo.upscalesoftware.com;
    set $MAGE_ROOT /var/www/magento2;
    include /var/www/magento2/vendor/upscale/magento2-swoole/devops/nginx.conf;
}
```

Magento Swoole Setup

Highlights:

- Swoole PHP extension installed
- Swoole integration module for Magento 2 installed

Deployment steps:

1. Install Swoole PHP extension
2. Install `upscale/magento2-swoole` Composer package
3. Install `Upscale_Swoole` Magento 2 module

MySQL Setup

Highlights:

- Timeouts increased
- Buffers increased
- SQL query caching enabled
- Provisioned IOPS storage

Configuration:

- RDS MySQL parameter group configuration:

```
connect_timeout = 60
net_read_timeout = 300
net_write_timeout = 600
innodb_lock_wait_timeout = 600
group_concat_max_len = 32768
log_bin_trust_function_creators = 1
query_cache_type = 1
query_cache_size = 67108864
max_allowed_packet = 268435456
max_heap_table_size = 536870912
tmp_table_size = 536870912
```

- RDS storage configuration:
  - Type: Provisioned IOPS (SSD)
  - Size: 100 GiB (required by Provisioned IOPS)
  - IOPS: 2,000

JMeter Setup

Highlights:

- OS tuning identical to web-server
- RAM of Java VM increased
- JMeter GUI mode disabled

Configuration:

- JMeter startup script `apache-jmeter-3.1/bin/jmeter`

```
HEAP="-Xms8g -Xmx30g"
NEW="-XX:NewSize=2g -XX:MaxNewSize=4g"
SERVER="-server -d64"
```

License