snowdog

Product Attribute Options

Performance analysis

Background

Site-Wide Analysis Tool

Reports > System Insights

SWAT Site Impact

Detail: Performance Limitation on number of Attribute Options

The following Attribute Options are above the **recommended limit of 100**.

The outstanding number of Attribute Options leads to an **increase of data retrieved** for each product on **all read and write operations** resulting in:

- Increase of SQL queries traffic and heavier JOIN operations affecting DB throughput
- Increase of Magento indexes size and full-text search index

SWAT Site Impact

These increases cause the following issues to occur on the site:

- **Response time on most storefront scenarios** related to products with containing an outstanding number of options in attributes will increase above performance targets.
- Product management operations in admin will significantly slow down and can lead to timeouts especially on scenarios related to attributes list and trees retrieval (including promo rules management).
- Product mass action functionality can be blocked.

Key things to check

- attribute definition
- storefront scenarios (PDP, PCP, Search, Cart, Checkout)
- php-spx profiling results
- query logs
- performance schema

Attribute definition

- SWAT reported attribute
- `select` frontend input
- `Magento\Eav\Model\Entity\Attribute\Source\Table` source model

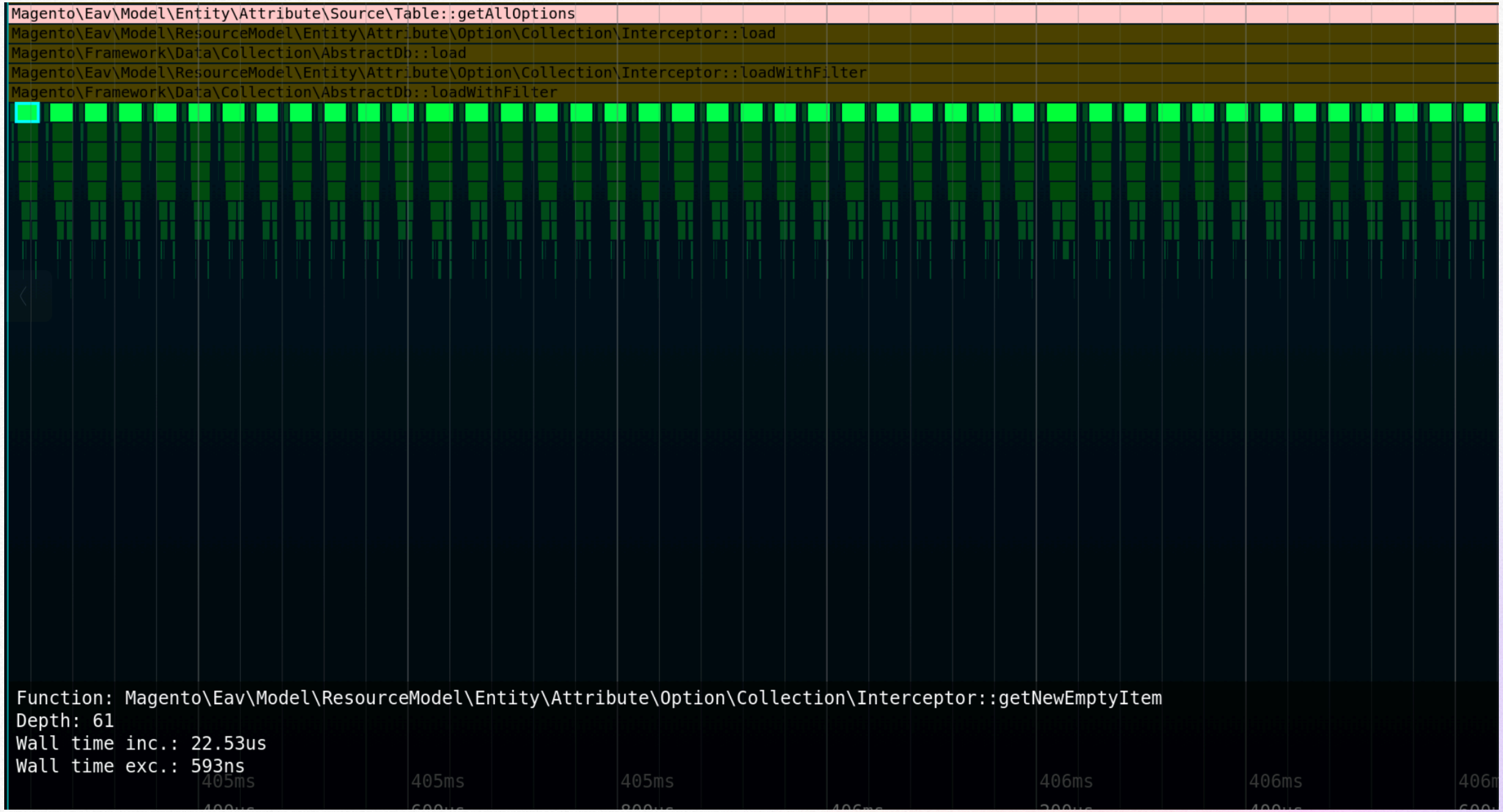
Storefront scenarios

- medium-large stores (30-150k SKUs), 3 stores
- data patch creating new `custom_size` attribute with 1k+ options
- php-spx profiling sessions showed no visible overhead on any page
- on the surface, increasing # of options did not drastically impact performance

php-spx insights

- the only "visible" overhead growth
- `` `Collection` `` processing of fetched query result rows

php-spx insights



```
Magento\Eav\Model\Entity\Attribute\Source\Table::getAllOptions
Magento\Eav\Model\ResourceModel\Entity\Attribute\Option\Collection\Interceptor::load
Magento\Framework\Data\Collection\AbstractDb::load
Magento\Eav\Model\ResourceModel\Entity\Attribute\Option\Collection\Interceptor::loadWithFilter
Magento\Framework\Data\Collection\AbstractDb::loadWithFilter
```

```
Function: Magento\Eav\Model\ResourceModel\Entity\Attribute\Option\Collection\Interceptor::getNewEmptyItem
Depth: 61
Wall time inc.: 22.53us
Wall time exc.: 593ns
```

Query logging

```
bin/magento dev:query-log:enable
```

Query #1

Get option text

```
SELECT `main_table`.*,
       `tdv`.`value`                                   AS `default_value`,
       `tsv`.`value`                                   AS `store_default_value`,
       IF(tsv.value_id > 0, tsv.value, tdv.value) AS `value`
FROM `eav_attribute_option` AS `main_table`
         INNER JOIN `eav_attribute_option_value` AS `tdv` ON tdv.option_id = main_table.option_id
         LEFT JOIN `eav_attribute_option_value` AS `tsv` ON tsv.option_id = main_table.option_id AND tsv.store_id = '1'
WHERE (`main_table`.`attribute_id` = '143')
  AND (`main_table`.`option_id` IN ('145'))
  AND (tdv.store_id = 0)
ORDER BY main_table.sort_order ASC, `value` ASC
```

Query #2

Get all options

```
SELECT `main_table`.*,
       `tdv`.`value`                                 AS `default_value`,
       `tsv`.`value`                                 AS `store_default_value`,
       IF(tsv.value_id > 0, tsv.value, tdv.value) AS `value`
FROM `eav_attribute_option` AS `main_table`
         INNER JOIN `eav_attribute_option_value` AS `tdv` ON tdv.option_id = main_table.option_id
         LEFT JOIN `eav_attribute_option_value` AS `tsv` ON tsv.option_id = main_table.option_id AND tsv.store_id = '1'
WHERE (`main_table`.`attribute_id` = '139')
  AND (tdv.store_id = 0)
ORDER BY main_table.sort_order ASC, `value` ASC
```

Other queries

- `vendor/magento/framework/EntityManager/Operation/Read/ReadAttributes.php`

```
SELECT `u`.*
FROM ((SELECT `t`.`value`, `t`.`attribute_id`, `t`.`store_id`
       FROM `catalog_product_entity_varchar` AS `t`
       WHERE (entity_id = '4781')
         AND (attribute_id IN
              (87, 88, 89, 135, 96, 104, 106, 73, 121, 124, 134, 122, 109, 110, 111, 84, 86, 100, 116, 117, 114, 126,
               127))
         AND (`store_id` IN ('1', 0)))
      UNION ALL
      (SELECT `t`.`value`, `t`.`attribute_id`, `t`.`store_id`
       FROM `catalog product entity int` AS `t`
```

Performance Schema

`.warden/warden-env.yml`

```
version: "3.5"
services:
  db:
    command:
      --performance_schema="on"
```

Performance Schema

- wait times in picoseconds (1e-12 [seconds])
- useful when checking how often tables are called, with what timings

Performance Schema - example 1

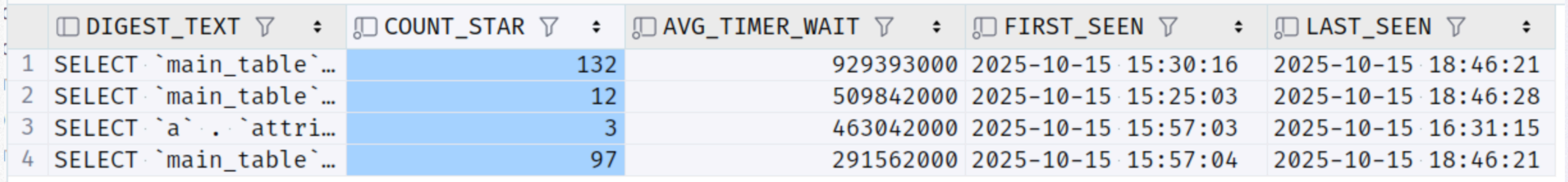
Check performance of specific tables

```
SELECT * FROM performance_schema.table_io_waits_summary_by_table
WHERE OBJECT_NAME LIKE "%eav_attribute_option%"
ORDER BY `AVG_TIMER_WAIT` DESC;
```

Performance Schema - example 2

Check performance of specific digested queries

```sql
SELECT * FROM performance_schema.events_statements_summary_by_digest
WHERE DIGEST_TEXT LIKE "%eav_attribute_option%"
ORDER BY `AVG_TIMER_WAIT` DESC;
```



	DIGEST_TEXT	COUNT_STAR	AVG_TIMER_WAIT	FIRST_SEEN	LAST_SEEN
1	SELECT `main_table`…	132	929393000	2025-10-15 15:30:16	2025-10-15 18:46:21
2	SELECT `main_table`…	12	509842000	2025-10-15 15:25:03	2025-10-15 18:46:28
3	SELECT `a` . `attri…	3	463042000	2025-10-15 15:57:03	2025-10-15 16:31:15
4	SELECT `main_table`…	97	291562000	2025-10-15 15:57:04	2025-10-15 18:46:21

```sql
SELECT `main_table`.*,
       `tdv`.`value`                                           AS `default_value`,
       `tsv`.`value`                                           AS `store_default_value`,
       IF(`tsv`.`value_id` > ?, `tsv`.`value`, `tdv`.`value`) AS `value`
FROM `eav_attribute_option` AS `main_table`
       INNER JOIN `eav_attribute_option_value` AS `tdv` ON `tdv`.`option_id` = `main_table`.`option_id`
       LEFT JOIN `eav_attribute_option_value` AS `tsv`
                 ON `tsv`.`option_id` = `main_table`.`option_id` AND `tsv`.`store_id` = ?
WHERE (`main_table`.`attribute_id` = ?)
  AND (`tdv`.`store_id` = ?)
ORDER BY `main_table`.`sort_order` ASC, `value` ASC
```

Performance Schema - something useful

Truncate all performance schema tables

```
CALL sys.ps_truncate_all_tables(FALSE);
```

SWAT Recommendation

- Leveraging different variation mechanisms: complex products, custom options as a source of product variations.
- Building specific product templates with targeting attributes and options, avoiding generalized product templates, and option containers.
- Maintaining a list of actual attribute options.
- Managing product info through external Product Management System (PMS).

Possible paths of optimization

Apart from admin work and maintaining "good attributes hygiene"

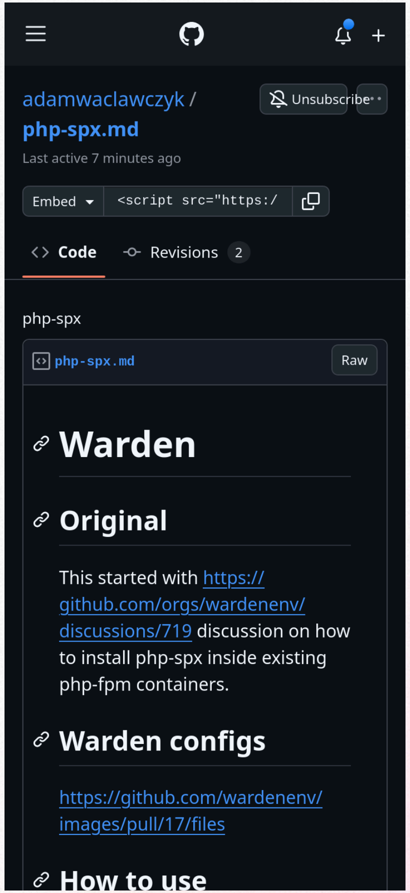
- custom source model classes
- compiled attribute / compiled attribute values to lessen the db load
- `Magento\Eav\Model\Entity\Attribute\Source\Table`
  - vs e.g. `\Magento\Eav\Model\Entity\Attribute\Source\Boolean`

Bonus

Guide to setting up php-spx

link

Link to above gist


adamwaclawczyk / php-spx.md

Last active 7 minutes ago

`<script src="https:/`

php-spx

php-spx.md

Warden

Original

This started with https://github.com/orgs/wardenenv/discussions/719 discussion on how to install php-spx inside existing php-fpm containers.

Warden configs

https://github.com/wardenenv/images/pull/17/files

How to use

Questions?

Feel free to ask, leave feedback below

slido q&a board

Link to presentation

Github pdf link

snowdog

Thank you for watching!

Adam Wacławczyk