MULTIGURA

Performance benchmark: MULTIGURA vs WPML

Front-end and admin page load, write operations and throughput on a bare WordPress site, with Elementor, Divi 5 and Avada, and in a WooCommerce shop. Each plugin runs on its own standalone single-site WordPress (no multisite) with identical content, next to a site without a multilingual plugin, all measured side by side in interleaved rounds.

Versions compared

Component	Version
PHP	**8.3.37** (PHP-FPM, 5 workers) + OPcache (`validate_timestamps=0` , 1 GB)
Web server	nginx 1.27.5
Database	MariaDB 11.8.2
WordPress	**7.1.5**, three standalone single-site installs (no multisite)
MULTIGURA	**2.0.16**
WPML	**Multilingual CMS 5.4.0** + String Translation 5.4.0 + Media Translation 5.4.0; WooCommerce Multilingual & Multicurrency 5.6.3 (WooCommerce scenario)
WooCommerce	11.3.1 (WooCommerce scenario, on all three sites)
Themes	Twenty Twenty-Five 1.4 · Astra 4.13.1 · Divi 5.19.1 · Avada 7.16.4 · Storefront 4.6.2
Builders	Elementor 4.4.1 + Elementor Pro 4.4.1 · Divi 5 (theme) · Avada Builder 3.16.4 + Avada Core 5.16.4
Machine	Docker Desktop on Apple Silicon (macOS), 8 vCPU, WordPress files bind-mounted

Date
2026-11-24

Requests measured
5,780 (errors: 0)

Scenarios
5: bare, Elementor, Divi 5, Avada, WooCommerce

1. Summary

+5 ms
MULTIGURA, JA page
vs. no plugin (+8%)

+46 ms
WPML, JA page
vs. no plugin (+77%)

52 / 52
faster than WPML
page × language × scenario

- **MULTIGURA adds little to front-end pages.** Averaged over all page types and the 5 scenarios, an English page takes **+4 ms** (+6%) more than the same site without a multilingual plugin, a Japanese page **+5 ms** (+8%). It was faster than WPML in **52 of 52** page, language and scenario combinations.
- **WPML** adds **+42 ms** (+71%) to English pages and **+46 ms** (+77%) to Japanese pages; on Japanese pages it needs **1.64×** the time of MULTIGURA. Both plugins store a translation as its own post, but WPML loads about 2 549 PHP files per front-end request (MULTIGURA 1 166, WordPress alone 1 124).
- **Translated pages cost MULTIGURA almost nothing extra.** A translation is rendered like the original: Japanese minus English is +0.8 ms on the content pages; in the shop Japanese pages take +0.6 ms more (product translation lookups), the cart +0.1 ms (section 4).
- **WooCommerce shop:** MULTIGURA adds +6 ms (EN) / +6 ms (JA) to shop pages, WPML with WooCommerce Multilingual +54 / +58 ms (section 4).
- **Under load:** with four concurrent clients MULTIGURA served **30.1** Japanese pages per second on average, WPML 20.4 (no plugin: 31.8 English pages per second).
- **Admin and writes:** admin screens average 91 ms with MULTIGURA (no plugin 85 ms, WPML 159 ms); write requests are dominated by WordPress itself (sections 5 and 6).

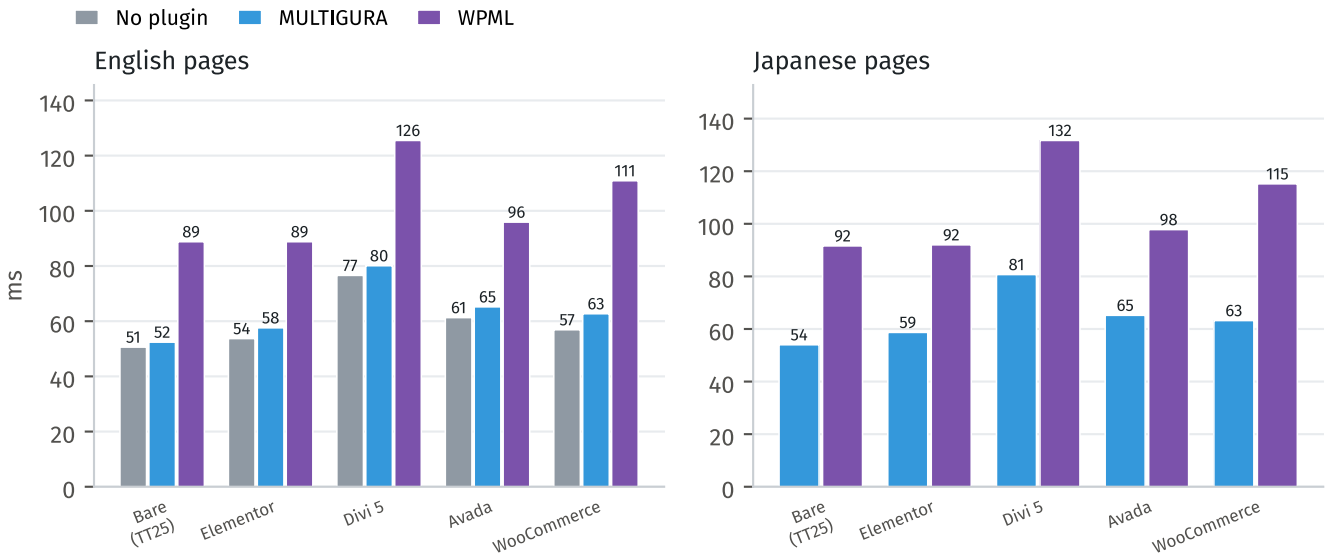

Figure 1. Mean of the median full-response times of the front-end page types per scenario (content scenarios: home, post, page, category, search; WooCommerce: home, shop, product, category, cart, search). Lower is better. Left: English (default language); right: the Japanese translation. “No plugin” has no translation, so it appears only on the left.

2. Test setup and method

Sites

Three **standalone single-site WordPress installs**: **No plugin** (baseline), **MULTIGURA** and **WPML**. They share the WordPress, plugin and theme files of one installation, but each has its own database tables (table prefix) and uploads folder; nothing is shared between them and none of them is a multisite network. The request header `X-Bench-Site` selects the site in `wp-config.php`, so all three are served by the same nginx and PHP-FPM workers and are measured in the same interleaved rounds. Each translation plugin is active only on its own site.

Content

- Same generator with a fixed random seed on every site: 300 posts (about 450 words, 1 or 2 categories, 3 tags, a featured image), 20 pages (5 with child pages), 12 images, 10 categories, 30 tags, a primary menu with 11 items;
- four landing pages with the same visible content (hero, six feature boxes, a grid of the 6 latest posts, a call to action) built with blocks, Elementor, Divi 5 and Avada Builder;
- WooCommerce: 8 product categories, 60 simple products (price, SKU, stock, image, 1 or 2 categories, short and long description; 6 featured, 12 on sale), the shop, cart, checkout and account pages, and a shop home page on Storefront's homepage template (product categories, recent, featured, popular and on-sale products);
- a Japanese (JA) translation of everything in each plugin's own data model. MULTIGURA: a separate post per language linked in its translations table, category names in term meta, a translated menu. WPML: a separate post, category, tag and menu per language linked in `icl_translations` (created through WPML's API). The Japanese text is a letter-for-letter transliteration of the English into katakana, so translated pages have the same length everywhere.

Two languages: English (default) and Japanese. Permalinks `/%postname%/`. MULTIGURA: posts, pages, categories (and products, product categories in the shop) translatable, media module on, default language at the root (no prefix), as on the other site. WPML: Japanese from its built-in language list (`ja`); language in directories, default language without a prefix.

Scenarios

Bare	Twenty Twenty-Five (block theme), front page built with blocks
Elementor	Astra + Elementor + Elementor Pro; Elementor front page (containers, heading, text, buttons, icon boxes, Recent Posts widget)
Divi 5	Divi theme; Divi 5 front page (heading, text, button, blurbs, Blog module)
Avada	Avada + Avada Builder + Avada Core; Avada front page (title, text, button, columns, Blog element)
WooCommerce	Storefront + WooCommerce on all sites (+ WooCommerce Multilingual on the WPML site); WooCommerce is off in the other scenarios

What was measured

- **Front-end** (anonymous visitor), English and Japanese: home, single post, child page, category archive, search; in the shop: shop home, shop page, product, product category, cart, product search.
- **Admin** (administrator): Dashboard, Posts and Pages lists, block editor for a post, Categories, Menus (classic themes), the builder's editor; in the shop: Products list, product editor, Product categories, Pages, WooCommerce settings.
- **Writes** over HTTP, as an editor does them: REST create, update and delete of a post, REST create of a category, saving the builder landing page and saving a Japanese translation (creating a JA post). In the shop: WooCommerce REST create, update and delete of a product, a JA product translation, and a guest "add to cart" on the English and the Japanese site (`?wc-ajax=add_to_cart` , new session each round). Plus a bulk test in one WP-CLI process (100 inserts, updates, translations, deletes).
- **Throughput:** 120 requests per URL with 4 concurrent clients (PHP-FPM has 5 workers) on home and post (shop: home and product), two passes in alternating order.

Procedure. Each URL was first requested 3 to 5 times (OPcache, builder CSS). Then 20 rounds; each round requests every URL of every site in a new random order, so drift on the machine affects all sites equally. Requests go from the PHP container to nginx over the Docker network with gzip and keep-alive. A temporary must-use plugin records PHP time from request start to shutdown, database queries, peak memory and included PHP files per request. Values are medians (p90 in the tables). OPcache with `validate_timestamps=0` as on a production server; no page cache, no persistent object cache, WP_DEBUG off, WP-Cron off on the test sites; MULTIGURA's own Memcached cache off. 5,780 measured requests, 0 errors.

Absolute times on Docker Desktop for macOS differ from a Linux server; the report compares the plugins with each other.

3. Front-end pages

Full response time (curl total: request to last byte), median of 20 interleaved rounds. The figure shows the Japanese pages of the multilingual plugins; the tables give both languages with p90, database queries and peak memory.



Figure 2. Japanese front-end pages per scenario, median full-response time in ms (lower is better).

Bare (TT25): front-end, ms (median)

Page	Lang	No plugin	MULTIGURA	WPML
Home	EN	**45** p90 51 · 31 q · 6.2 MB	**46** p90 55 · 39 q · 6.5 MB	**78** p90 90 · 60 q · 10.6 MB
Home	JA	–	**44** p90 55 · 43 q · 6.6 MB	**84** p90 93 · 61 q · 12.1 MB
Post	EN	**49** p90 57 · 38 q · 6.2 MB	**51** p90 59 · 46 q · 6.6 MB	**82** p90 95 · 73 q · 10.7 MB
Post	JA	–	**56** p90 58 · 46 q · 6.6 MB	**90** p90 103 · 82 q · 12.2 MB
Page	EN	**40** p90 45 · 25 q · 6.2 MB	**44** p90 51 · 31 q · 6.6 MB	**74** p90 82 · 51 q · 10.5 MB
Page	JA	–	**46** p90 51 · 35 q · 6.6 MB	**73** p90 84 · 56 q · 11.9 MB
Category	EN	**56** p90 68 · 33 q · 6.3 MB	**56** p90 68 · 40 q · 6.7 MB	**96** p90 109 · 59 q · 10.8 MB
Category	JA	–	**58** p90 65 · 40 q · 6.8 MB	**99** p90 110 · 66 q · 12.4 MB
Search	EN	**63** p90 70 · 34 q · 6.3 MB	**65** p90 72 · 43 q · 6.7 MB	**114** p90 132 · 57 q · 10.8 MB
Search	JA	–	**67** p90 75 · 48 q · 6.8 MB	**111** p90 140 · 66 q · 12.4 MB

Bold: median full-response time; then p90, database queries and PHP peak memory of the request.

Elementor: front-end, ms (median)

Page	Lang	No plugin	MULTIGURA	WPML
Home	EN	**63** p90 67 · 110 q · 11.5 MB	**67** p90 68 · 119 q · 12.0 MB	**94** p90 100 · 146 q · 16.9 MB
Home	JA	–	**68** p90 69 · 123 q · 12.1 MB	**97** p90 101 · 152 q · 18.5 MB
Post	EN	**49** p90 57 · 100 q · 11.1 MB	**53** p90 55 · 111 q · 11.5 MB	**80** p90 82 · 143 q · 16.2 MB
Post	JA	–	**54** p90 56 · 112 q · 11.5 MB	**84** p90 87 · 151 q · 17.6 MB
Page	EN	**48** p90 49 · 93 q · 11.1 MB	**52** p90 53 · 103 q · 11.5 MB	**78** p90 83 · 127 q · 16.2 MB
Page	JA	–	**52** p90 53 · 106 q · 11.5 MB	**81** p90 85 · 136 q · 17.6 MB
Category	EN	**53** p90 55 · 103 q · 11.2 MB	**57** p90 62 · 114 q · 11.7 MB	**85** p90 87 · 139 q · 16.1 MB
Category	JA	–	**58** p90 59 · 115 q · 11.7 MB	**91** p90 98 · 145 q · 17.9 MB
Search	EN	**56** p90 58 · 98 q · 11.2 MB	**60** p90 62 · 109 q · 11.6 MB	**106** p90 109 · 126 q · 16.2 MB
Search	JA	–	**62** p90 65 · 111 q · 11.7 MB	**106** p90 112 · 136 q · 17.9 MB

Bold: median full-response time; then p90, database queries and PHP peak memory of the request.

Divi 5: front-end, ms (median)

Page	Lang	No plugin	MULTIGURA	WPML
Home	EN	**84** p90 88 · 75 q · 13.2 MB	**88** p90 89 · 83 q · 13.6 MB	**113** p90 115 · 124 q · 17.9 MB
Home	JA	–	**88** p90 90 · 87 q · 13.7 MB	**121** p90 122 · 127 q · 21.1 MB
Post	EN	**71** p90 73 · 63 q · 13.2 MB	**74** p90 76 · 71 q · 13.6 MB	**118** p90 123 · 295 q · 18.0 MB
Post	JA	–	**75** p90 79 · 72 q · 13.6 MB	**125** p90 128 · 302 q · 21.0 MB
Page	EN	**71** p90 72 · 61 q · 13.2 MB	**74** p90 77 · 70 q · 13.6 MB	**117** p90 119 · 284 q · 18.0 MB
Page	JA	–	**74** p90 77 · 73 q · 13.6 MB	**124** p90 126 · 296 q · 21.0 MB
Category	EN	**78** p90 79 · 86 q · 13.3 MB	**83** p90 84 · 95 q · 13.7 MB	**133** p90 135 · 324 q · 18.2 MB
Category	JA	–	**83** p90 85 · 96 q · 13.7 MB	**140** p90 142 · 333 q · 21.1 MB
Search	EN	**79** p90 82 · 73 q · 13.3 MB	**82** p90 83 · 82 q · 13.7 MB	**146** p90 151 · 297 q · 18.1 MB
Search	JA	–	**83** p90 86 · 84 q · 13.7 MB	**149** p90 152 · 304 q · 21.1 MB

Bold: median full-response time; then p90, database queries and PHP peak memory of the request.

Avada: front-end, ms (median)

Page	Lang	No plugin	MULTIGURA	WPML
Home	EN	**63** p90 64 · 61 q · 8.9 MB	**68** p90 70 · 69 q · 9.4 MB	**90** p90 91 · 93 q · 14.0 MB
Home	JA	–	**67** p90 70 · 73 q · 9.5 MB	**92** p90 95 · 102 q · 15.7 MB
Post	EN	**60** p90 63 · 68 q · 8.8 MB	**64** p90 65 · 78 q · 9.2 MB	**87** p90 88 · 119 q · 14.0 MB
Post	JA	–	**63** p90 66 · 79 q · 9.3 MB	**90** p90 92 · 132 q · 15.7 MB
Page	EN	**54** p90 56 · 46 q · 8.7 MB	**57** p90 58 · 54 q · 9.1 MB	**78** p90 80 · 79 q · 13.8 MB
Page	JA	–	**57** p90 59 · 57 q · 9.1 MB	**81** p90 82 · 87 q · 15.4 MB
Category	EN	**60** p90 62 · 59 q · 8.9 MB	**65** p90 71 · 70 q · 9.3 MB	**88** p90 90 · 94 q · 14.0 MB
Category	JA	–	**65** p90 67 · 71 q · 9.4 MB	**91** p90 93 · 100 q · 15.7 MB
Search	EN	**68** p90 70 · 59 q · 8.9 MB	**72** p90 76 · 70 q · 9.4 MB	**137** p90 140 · 277 q · 14.5 MB
Search	JA	–	**73** p90 75 · 72 q · 9.6 MB	**135** p90 138 · 288 q · 16.3 MB

Bold: median full-response time; then p90, database queries and PHP peak memory of the request.

4. WooCommerce shop

The shop scenario runs Storefront with WooCommerce on every site; the WPML site also runs WooCommerce Multilingual, as a WPML shop would. Mean over the six shop pages, compared with the shop without a multilingual plugin: MULTIGURA +6 ms EN / +6 ms JA, WPML +54 / +58 ms. The cart is the heaviest page for every site: WooCommerce renders it as a block page with its settings and a product list inlined.

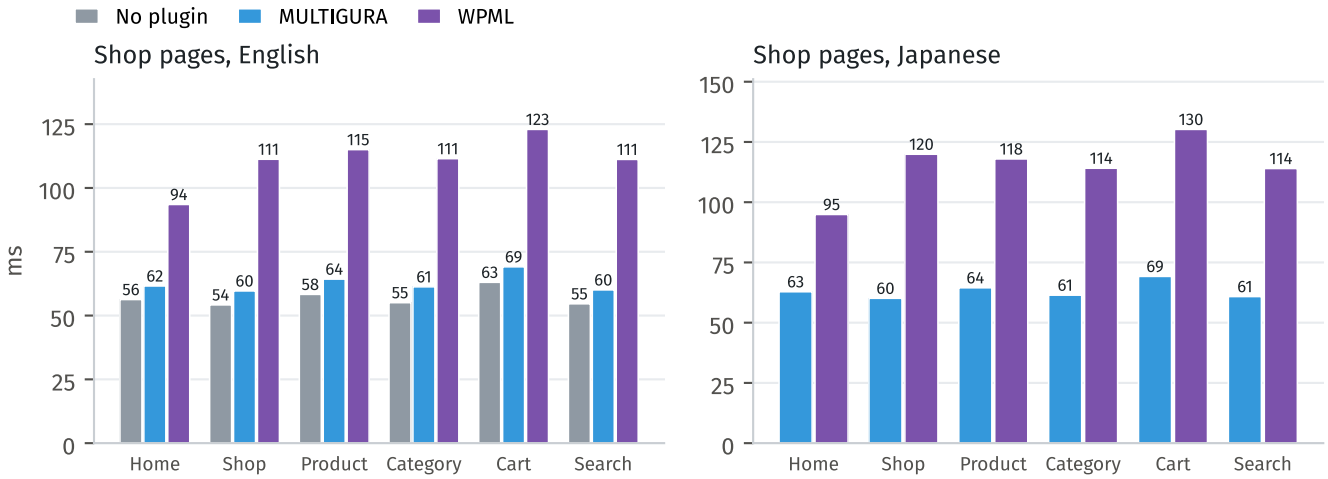

Figure 3. Shop pages, median full-response time in ms (lower is better). Left: English; right: Japanese.

WooCommerce: front-end, ms (median)

Page	Lang	No plugin	MULTIGURA	WPML
Home	EN	**56** p90 58 · 108 q · 11.1 MB	**62** p90 65 · 121 q · 11.7 MB	**94** p90 95 · 159 q · 17.0 MB
Home	JA	–	**63** p90 64 · 131 q · 11.9 MB	**95** p90 97 · 171 q · 19.0 MB
Shop	EN	**54** p90 56 · 80 q · 11.1 MB	**60** p90 61 · 90 q · 11.8 MB	**111** p90 114 · 314 q · 17.6 MB
Shop	JA	–	**60** p90 61 · 94 q · 11.9 MB	**120** p90 124 · 367 q · 19.9 MB
Product	EN	**58** p90 60 · 100 q · 11.1 MB	**64** p90 66 · 118 q · 11.6 MB	**115** p90 118 · 344 q · 17.5 MB
Product	JA	–	**64** p90 67 · 121 q · 11.7 MB	**118** p90 120 · 353 q · 19.6 MB
Category	EN	**55** p90 56 · 84 q · 11.2 MB	**61** p90 69 · 96 q · 11.8 MB	**111** p90 113 · 321 q · 17.6 MB
Category	JA	–	**61** p90 66 · 100 q · 11.9 MB	**114** p90 115 · 323 q · 19.8 MB
Cart	EN	**63** p90 64 · 98 q · 12.1 MB	**69** p90 71 · 108 q · 12.6 MB	**123** p90 127 · 345 q · 18.9 MB
Cart	JA	–	**69** p90 71 · 115 q · 12.8 MB	**130** p90 134 · 350 q · 21.9 MB
Search	EN	**55** p90 55 · 79 q · 11.2 MB	**60** p90 67 · 89 q · 11.7 MB	**111** p90 114 · 312 q · 17.6 MB
Search	JA	–	**61** p90 62 · 94 q · 11.9 MB	**114** p90 118 · 319 q · 19.8 MB

Bold: median full-response time; then p90, database queries and PHP peak memory of the request.

Shop writes

Server time (PHP), median of 15 rounds. Product create, update and delete (WooCommerce REST API, mean): no plugin 44 ms, MULTIGURA 49 ms, WPML 112 ms. Guest add to cart on the Japanese site: MULTIGURA 30 ms, WPML 52 ms (no plugin, English: 24 ms).

Operation	No plugin	MULTIGURA	WPML
Create product	**50** 253 q	**55** 276 q	**133** 585 q
Update product	**38** 125 q	**41** 142 q	**92** 233 q
Save JA product translation	–	**52** 256 q	**132** 545 q
Add to cart (EN, guest)	**24** 51 q	**26** 57 q	**54** 94 q
Add to cart (JA, guest)	–	**30** 62 q	**52** 84 q
Delete product	**45** 221 q	**50** 249 q	**112** 450 q

Shop admin screens (PHP time, ms)

Screen	No plugin	MULTIGURA	WPML
Dashboard	**74** 118 q	**76** 118 q	**157** 191 q
Pages list	**81** 117 q	**89** 120 q	**184** 200 q
Products list	**114** 136 q	**127** 138 q	**198** 206 q
Edit product	**102** 139 q	**107** 154 q	**178** 219 q
Product categories	**93** 116 q	**97** 114 q	**165** 190 q
WooCommerce settings	**86** 111 q	**89** 108 q	**150** 169 q

5. Admin screens

Server time (PHP) of admin screens, mean over the screens of all scenarios: no plugin 85 ms, MULTIGURA 91 ms, WPML 159 ms. Builder editors are dominated by the builder itself. WPML loads about 3 204 PHP files per admin request (MULTIGURA 1 149, WordPress alone 1 105). The WooCommerce screens are in section 4.

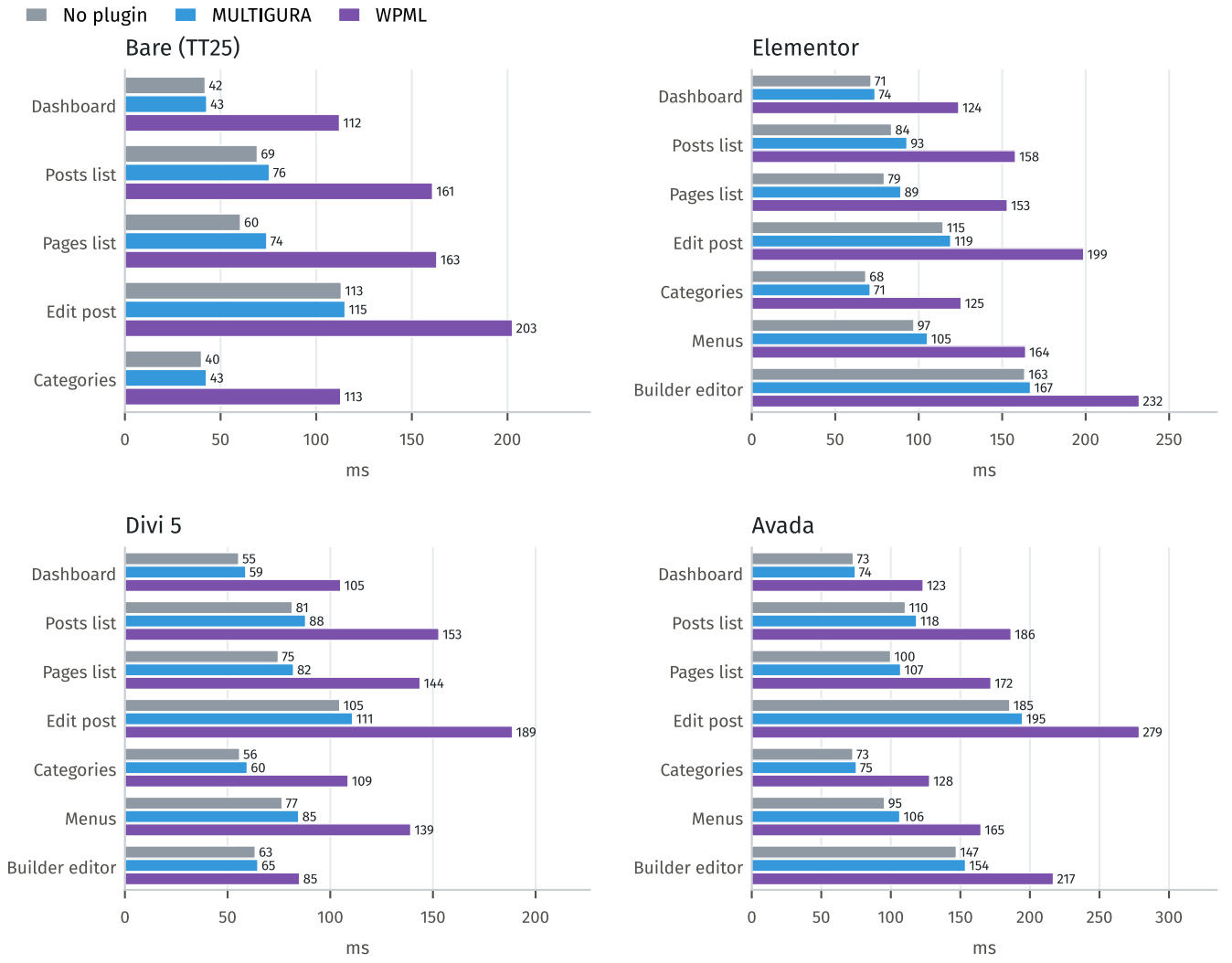

Figure 4. Admin screens, server (PHP) time in ms, median (lower is better).

Scenario	Screen	No plugin	MULTIGURA	WPML
Bare (TT25)	Dashboard	**42** 39 q	**43** 43 q	**112** 93 q
Bare (TT25)	Posts list	**69** 44 q	**76** 48 q	**161** 114 q
Bare (TT25)	Pages list	**60** 36 q	**74** 43 q	**163** 107 q
Bare (TT25)	Edit post	**113** 56 q	**115** 76 q	**203** 118 q
Bare (TT25)	Categories	**40** 29 q	**43** 31 q	**113** 111 q
Elementor	Dashboard	**71** 116 q	**74** 122 q	**124** 174 q
Elementor	Posts list	**84** 115 q	**93** 141 q	**158** 188 q
Elementor	Pages list	**79** 107 q	**89** 136 q	**153** 181 q
Elementor	Edit post	**115** 138 q	**119** 159 q	**199** 202 q
Elementor	Categories	**68** 100 q	**71** 104 q	**125** 185 q

Scenario	Screen	No plugin	MULTIGURA	WPML
Elementor	Menus	**97** 217 q	**105** 227 q	**164** 324 q
Elementor	Builder editor	**163** 140 q	**167** 147 q	**232** 188 q
Divi 5	Dashboard	**55** 55 q	**59** 61 q	**105** 112 q
Divi 5	Posts list	**81** 59 q	**88** 63 q	**153** 132 q
Divi 5	Pages list	**75** 51 q	**82** 57 q	**144** 122 q
Divi 5	Edit post	**105** 83 q	**111** 103 q	**189** 148 q
Divi 5	Categories	**56** 44 q	**60** 48 q	**109** 129 q
Divi 5	Menus	**77** 157 q	**85** 163 q	**139** 258 q
Divi 5	Builder editor	**63** 50 q	**65** 52 q	**85** 74 q
Avada	Dashboard	**73** 71 q	**74** 75 q	**123** 126 q
Avada	Posts list	**110** 98 q	**118** 103 q	**186** 171 q
Avada	Pages list	**100** 66 q	**107** 74 q	**172** 140 q
Avada	Edit post	**185** 131 q	**195** 151 q	**279** 200 q
Avada	Categories	**73** 63 q	**75** 65 q	**128** 146 q
Avada	Menus	**95** 210 q	**106** 254 q	**165** 322 q
Avada	Builder editor	**147** 94 q	**154** 98 q	**217** 128 q

6. Write operations

Server time (PHP) of write requests, median of 15 rounds. Mean over create, update and delete post, create category and builder save: no plugin 90 ms, MULTIGURA 93 ms, WPML 161 ms. Most of the time is WordPress core (revisions, term counts, caches). The WooCommerce writes are in section 4.

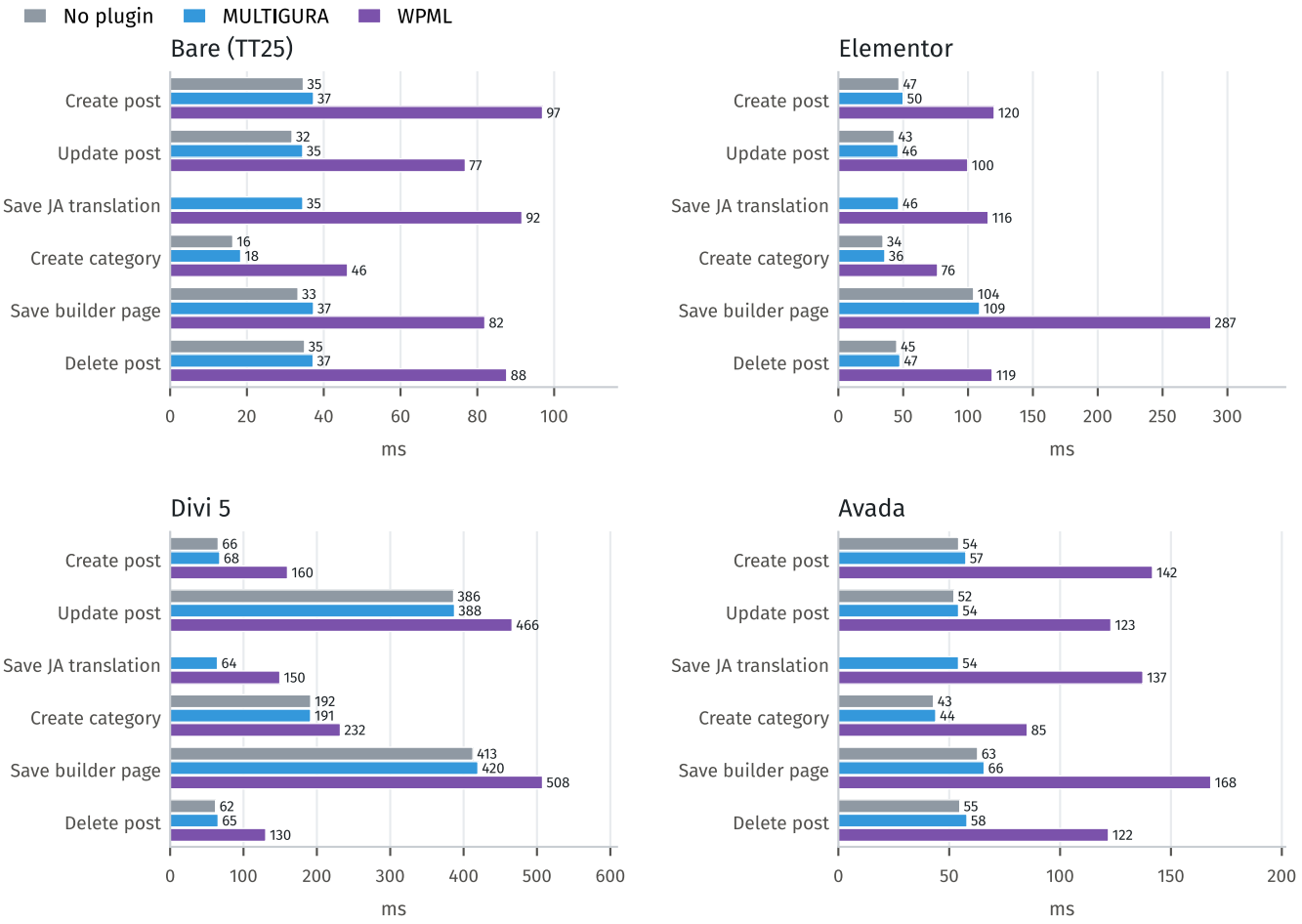

Figure 5. Write requests, server (PHP) time in ms, median (lower is better).

Scenario	Operation	No plugin	MULTIGURA	WPML
Bare (TT25)	Create post	**35** 103 q	**37** 119 q	**97** 348 q
Bare (TT25)	Update post	**32** 51 q	**35** 65 q	**77** 158 q
Bare (TT25)	Save JA translation	–	**35** 69 q	**92** 292 q
Bare (TT25)	Create category	**16** 26 q	**18** 29 q	**46** 72 q
Bare (TT25)	Save builder page	**33** 43 q	**37** 62 q	**82** 163 q
Bare (TT25)	Delete post	**35** 82 q	**37** 97 q	**88** 301 q
Elementor	Create post	**47** 159 q	**50** 175 q	**120** 409 q
Elementor	Update post	**43** 107 q	**46** 121 q	**100** 217 q
Elementor	Save JA translation	–	**46** 125 q	**116** 353 q
Elementor	Create category	**34** 84 q	**36** 87 q	**76** 131 q
Elementor	Save builder page	**104** 269 q	**109** 310 q	**287** 589 q
Elementor	Delete post	**45** 143 q	**47** 161 q	**119** 368 q

Scenario	Operation	No plugin	MULTIGURA	WPML
Divi 5	Create post	**66** 149 q	**68** 165 q	**160** 482 q
Divi 5	Update post	**386** 88 q	**388** 102 q	**466** 295 q
Divi 5	Save JA translation	–	**64** 115 q	**150** 426 q
Divi 5	Create category	**192** 50 q	**191** 53 q	**232** 98 q
Divi 5	Save builder page	**413** 97 q	**420** 117 q	**508** 355 q
Divi 5	Delete post	**62** 113 q	**65** 128 q	**130** 340 q
Avada	Create post	**54** 125 q	**57** 141 q	**142** 453 q
Avada	Update post	**52** 76 q	**54** 90 q	**123** 272 q
Avada	Save JA translation	–	**54** 91 q	**137** 397 q
Avada	Create category	**43** 44 q	**44** 47 q	**85** 94 q
Avada	Save builder page	**63** 79 q	**66** 95 q	**168** 400 q
Avada	Delete post	**55** 100 q	**58** 115 q	**122** 321 q

Bulk writes (one WP-CLI process, per item)

100 posts inserted, updated, translated (MULTIGURA, WPML: a JA post through the plugin's API) and deleted in one PHP process; time and queries per item.

Plugin	Insert	Update	Add JA translation	Delete
No plugin	3.2 ms · 37 q	2.4 ms · 23 q	–	3.0 ms · 37 q
MULTIGURA	3.9 ms · 46 q	3.6 ms · 30 q	4.9 ms · 49 q	3.6 ms · 38 q
WPML	8.8 ms · 86 q	7.1 ms · 64 q	9.7 ms · 96 q	5.1 ms · 59 q

7. Throughput

Requests per second with 4 concurrent clients (higher is better). The multilingual plugins are shown on the Japanese page, the baseline on its English page.

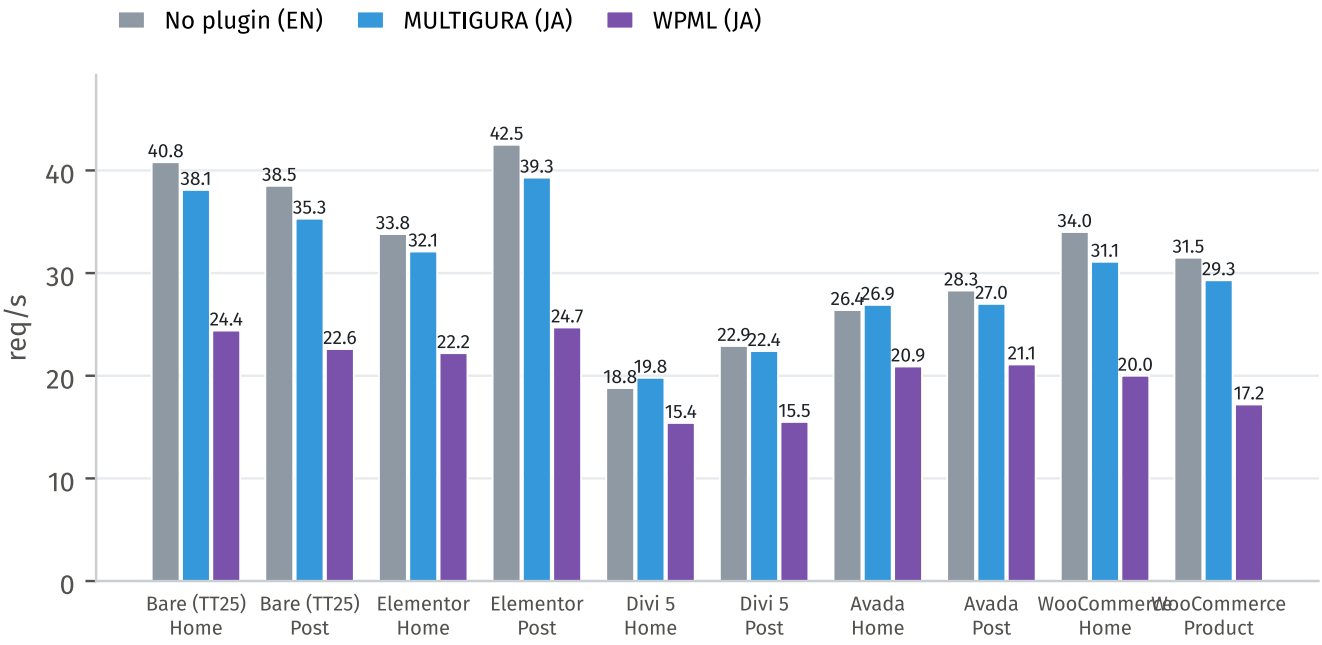

Figure 6. Requests per second with 4 concurrent clients (higher is better).

Scenario	Page	Lang	No plugin	MULTIGURA	WPML
Bare (TT25)	Home	EN	**40.8** med 76 · p95 92 ms	**38.4** med 82 · p95 93 ms	**25.1** med 126 · p95 140 ms
Bare (TT25)	Home	JA	–	**38.1** med 80 · p95 97 ms	**24.4** med 130 · p95 147 ms
Bare (TT25)	Post	EN	**38.5** med 80 · p95 94 ms	**37.0** med 84 · p95 95 ms	**23.6** med 134 · p95 148 ms
Bare (TT25)	Post	JA	–	**35.3** med 87 · p95 126 ms	**22.6** med 138 · p95 153 ms
Elementor	Home	EN	**33.8** med 106 · p95 118 ms	**32.0** med 111 · p95 123 ms	**23.3** med 153 · p95 169 ms
Elementor	Home	JA	–	**32.1** med 112 · p95 122 ms	**22.2** med 159 · p95 175 ms
Elementor	Post	EN	**42.5** med 74 · p95 84 ms	**40.4** med 78 · p95 90 ms	**25.4** med 126 · p95 136 ms
Elementor	Post	JA	–	**39.3** med 81 · p95 90 ms	**24.7** med 131 · p95 142 ms
Divi 5	Home	EN	**18.8** med 151 · p95 170 ms	**18.5** med 156 · p95 186 ms	**15.2** med 202 · p95 219 ms
Divi 5	Home	JA	–	**19.8** med 157 · p95 174 ms	**15.4** med 215 · p95 234 ms
Divi 5	Post	EN	**22.9** med 134 · p95 150 ms	**22.4** med 138 · p95 152 ms	**15.9** med 205 · p95 226 ms
Divi 5	Post	JA	–	**22.4** med 137 · p95 150 ms	**15.5** med 214 · p95 231 ms
Avada	Home	EN	**26.4** med 116 · p95 155 ms	**26.4** med 115 · p95 135 ms	**20.7** med 162 · p95 193 ms
Avada	Home	JA	–	**26.9** med 113 · p95 140 ms	**20.9** med 155 · p95 184 ms
Avada	Post	EN	**28.3** med 108 · p95 134 ms	**27.4** med 114 · p95 139 ms	**21.3** med 152 · p95 186 ms
Avada	Post	JA	–	**27.0** med 118 · p95 139 ms	**21.1** med 148 · p95 180 ms
WooCommerce	Home	EN	**34.0** med 88 · p95 105 ms	**31.2** med 97 · p95 105 ms	**20.5** med 150 · p95 163 ms
WooCommerce	Home	JA	–	**31.1** med 98 · p95 106 ms	**20.0** med 155 · p95 166 ms
WooCommerce	Product	EN	**31.5** med 97 · p95 106 ms	**28.8** med 106 · p95 115 ms	**17.4** med 184 · p95 194 ms

Scenario	Page	Lang	No plugin	MULTIGURA	WPML
WooCommerce	Product	JA	–	**29.3** med 104 · p95 115 ms	**17.2** med 184 · p95 198 ms

8. Code and queries per request

What a request loads, mean of the medians over all front-end pages of all scenarios (admin: over all admin screens). PHP files counts every included file (WordPress core, theme, builder and plugins together), so the difference to “No plugin” is the plugin’s own share.

Plugin	PHP files, EN	PHP files, JA	PHP files, admin	Queries, EN	Queries, JA	Memory EN, MB	Memory JA, MB
No plugin	1 124	–	1 105	72	–	10.2	–
MULTIGURA	1 166	1 166	1 149	82	85	10.7	10.7
WPML	2 549	2 557	3 204	183	192	15.5	17.5

9. Findings

- **Overall.** MULTIGURA was faster than WPML in 52 of 52 combinations.
- **Japanese pages:** on average MULTIGURA adds +5 ms and WPML +46 ms over the site without a plugin; WPML needs 1.64× the time of MULTIGURA.
- **Search:** MULTIGURA adds +3 ms to an English search page over the baseline (content scenarios); home, post, page and category add +3 ms. The search query reads only the posts of the current language.
- **WooCommerce:** MULTIGURA adds +6 ms (EN) / +6 ms (JA) on shop pages, WPML with WooCommerce Multilingual +54 / +58 ms. The cart: MULTIGURA 69 / 69 ms, WPML 123 / 130 ms (EN / JA).
- **Queries:** MULTIGURA runs about 10 more queries per English page than WordPress alone (Japanese 13), WPML about 111 (Japanese 121).
- **Code footprint:** 42 extra PHP files per front-end request with MULTIGURA, 1 425 with WPML.

10. Limits

- One machine, Docker Desktop on macOS: absolute numbers differ from a production Linux server; differences between the plugins are reliable to a few milliseconds (interleaved rounds, medians).
- Page caching hides most of these differences for anonymous visitors; they matter for cache misses, logged-in users, the cart and checkout, and search.
- Two languages were measured.
- The shop has simple products only (no variations) and an empty cart for page loads; checkout and payment were not measured.
- Builder editors are JavaScript apps; only their server response is measured.
- Each plugin ran with its defaults plus the settings listed in section 2; WPML's Advanced Translation Editor was not connected; MULTIGURA's Memcached cache was off.
- The measurement was done by the MULTIGURA team. The scripts and raw results are available on request.