MULTIGURA

Performance benchmark: MULTIGURA vs TranslatePress

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**
TranslatePress	**3.4** (TranslatePress Multilingual) + **Developer 1.9.0** (SEO Pack on)
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 (+9%)

+67 ms
TranslatePress, JA page
vs. no plugin (+114%)

52 / 52
faster than TranslatePress
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** (+7%) more than the same site without a multilingual plugin, a Japanese page **+5 ms** (+9%). It was faster than TranslatePress in **52 of 52** page, language and scenario combinations.
- **TranslatePress** adds **+21 ms** (+37%) to English pages and **+67 ms** (+114%) to Japanese pages. It translates the finished page: it buffers the HTML, parses it and looks every string up in its dictionary, so its Japanese pages are clearly slower than its English ones (1.97× the time of MULTIGURA on Japanese pages).
- **Translated pages cost MULTIGURA almost nothing extra.** A translation is its own post and is rendered like the original: Japanese minus English is +0.8 ms on the content pages; in the shop Japanese pages take +0.8 ms more (product translation lookups), the cart −0.1 ms (section 4).
- **WooCommerce shop:** MULTIGURA adds +6 ms (EN) / +7 ms (JA) to shop pages, TranslatePress +46 / +107 ms; the cart takes 72 ms with MULTIGURA and 296 ms with TranslatePress in Japanese (section 4).
- **Under load:** with four concurrent clients MULTIGURA served **30.0** Japanese pages per second on average, TranslatePress 16.8 (no plugin: 31.4 English pages per second).
- **Admin and writes:** admin screens average 95 ms with MULTIGURA (no plugin 89 ms, TranslatePress 98 ms); write requests are dominated by WordPress itself (sections 5 and 6).

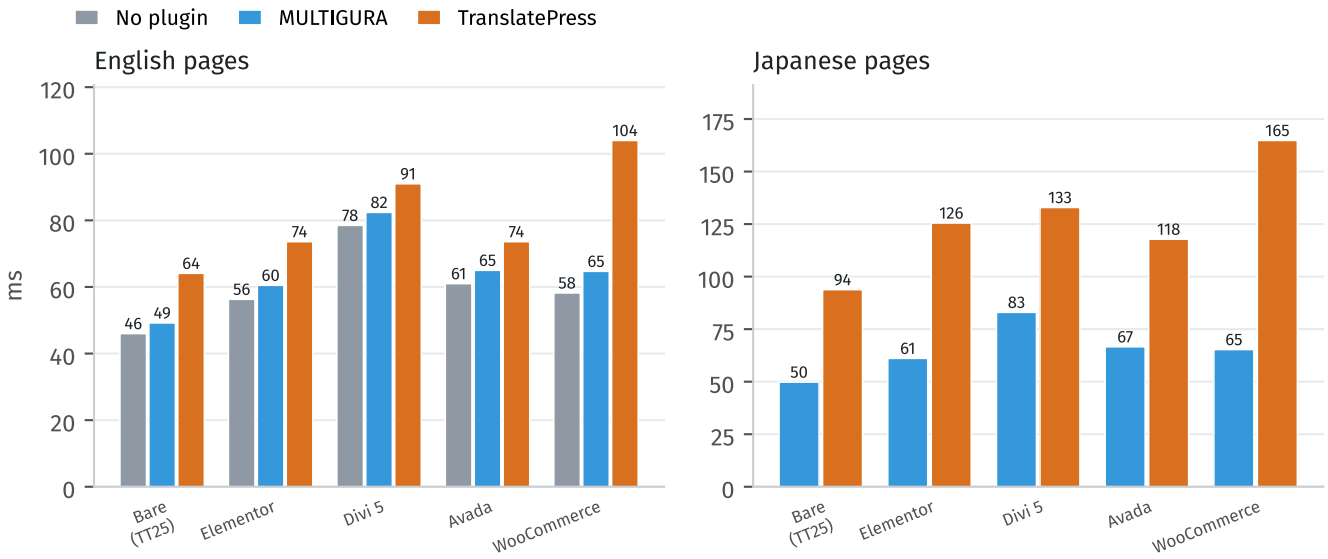

Figure 1. Mean of the median full-response times of the front-end page types per scenario (content scenarios: home, post, page, category, search; WooCommerce: home, shop, product, category, cart, search). Lower is better. Left: English (default language); right: the Japanese translation. “No plugin” 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 **TranslatePress**. 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. TranslatePress: one copy of the content plus a filled Japanese dictionary (every string on the measured pages translated, status "human reviewed"). 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, as in the other benchmarks. TranslatePress runs with its Developer add-ons and the SEO Pack active (translatable URL slugs and SEO metadata); Japanese URL slugs stay the same as the English ones, as on the other sites. 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.

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 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 (MULTIGURA: creating a JA post; TranslatePress: saving 15 strings through its editor AJAX). 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, TranslatePress dictionary). 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.

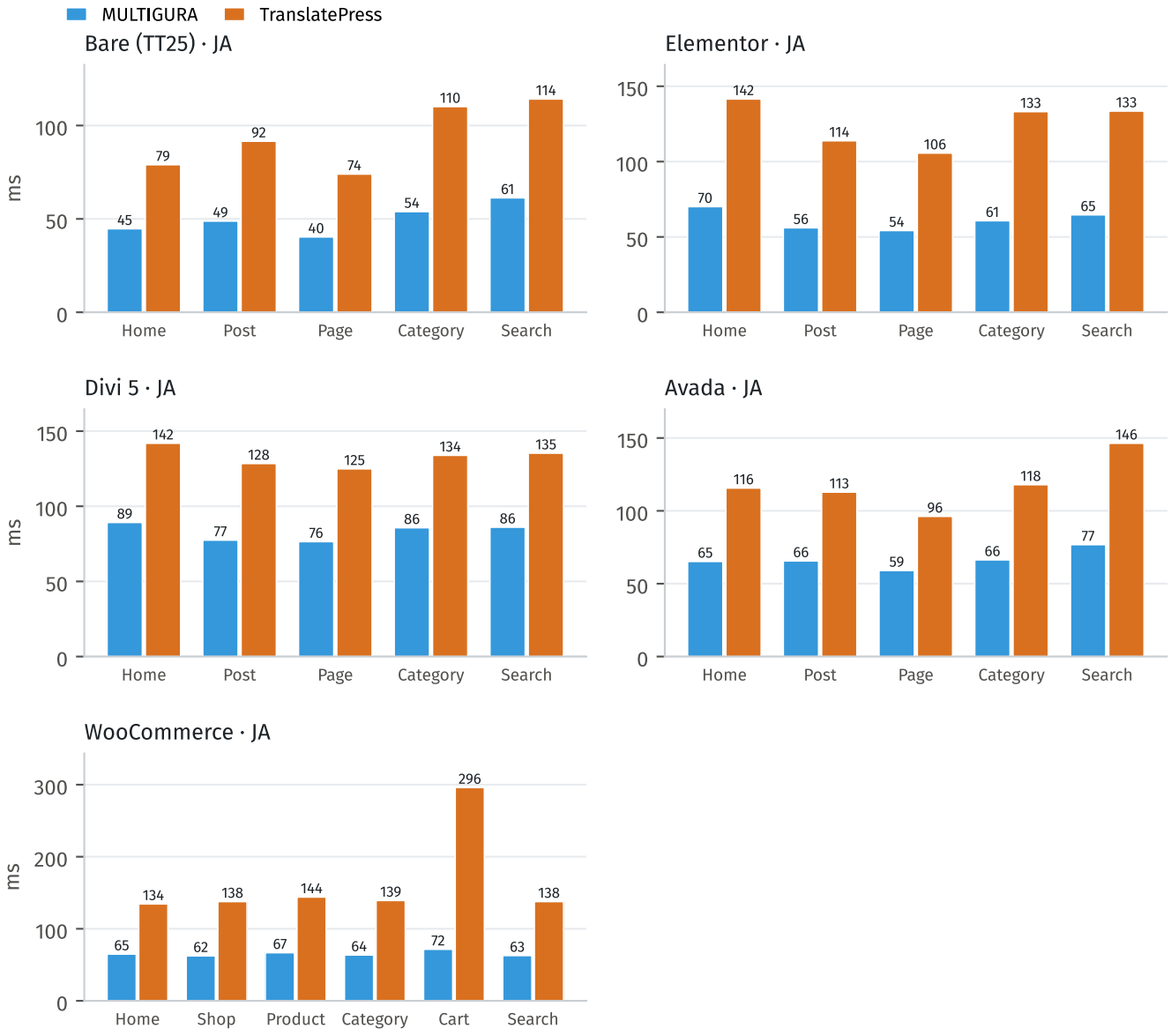

Figure 2. Japanese front-end pages per scenario, median full-response time in ms (lower is better).

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

Page	Lang	No plugin	MULTIGURA	TranslatePress
Home	EN	**41** p90 44 · 31 q · 6.2 MB	**44** p90 47 · 39 q · 6.6 MB	**59** p90 63 · 48 q · 7.6 MB
Home	JA	–	**45** p90 47 · 43 q · 6.6 MB	**79** p90 83 · 100 q · 9.9 MB
Post	EN	**45** p90 48 · 38 q · 6.2 MB	**47** p90 52 · 46 q · 6.6 MB	**63** p90 66 · 54 q · 7.8 MB
Post	JA	–	**49** p90 51 · 46 q · 6.6 MB	**92** p90 99 · 143 q · 10.5 MB
Page	EN	**39** p90 41 · 25 q · 6.2 MB	**41** p90 44 · 31 q · 6.6 MB	**56** p90 59 · 45 q · 7.5 MB
Page	JA	–	**40** p90 44 · 35 q · 6.6 MB	**74** p90 79 · 101 q · 9.5 MB
Category	EN	**51** p90 54 · 33 q · 6.3 MB	**54** p90 58 · 40 q · 6.7 MB	**70** p90 74 · 49 q · 7.8 MB
Category	JA	–	**54** p90 59 · 40 q · 6.8 MB	**110** p90 115 · 139 q · 12.5 MB
Search	EN	**55** p90 58 · 34 q · 6.3 MB	**59** p90 61 · 46 q · 6.7 MB	**72** p90 77 · 47 q · 7.8 MB
Search	JA	–	**61** p90 66 · 49 q · 6.8 MB	**114** p90 118 · 129 q · 12.7 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	TranslatePress
Home	EN	**65** p90 66 · 110 q · 11.5 MB	**69** p90 72 · 119 q · 12.0 MB	**94** p90 98 · 130 q · 13.9 MB
Home	JA	–	**70** p90 77 · 123 q · 12.1 MB	**142** p90 153 · 318 q · 17.6 MB
Post	EN	**52** p90 54 · 100 q · 11.1 MB	**56** p90 58 · 111 q · 11.5 MB	**67** p90 70 · 120 q · 12.4 MB
Post	JA	–	**56** p90 62 · 112 q · 11.5 MB	**114** p90 121 · 325 q · 15.5 MB
Page	EN	**50** p90 53 · 93 q · 11.1 MB	**54** p90 60 · 103 q · 11.5 MB	**62** p90 65 · 113 q · 12.4 MB
Page	JA	–	**54** p90 58 · 106 q · 11.5 MB	**106** p90 110 · 321 q · 14.7 MB
Category	EN	**57** p90 60 · 99 q · 11.2 MB	**62** p90 63 · 109 q · 11.6 MB	**72** p90 75 · 119 q · 12.6 MB
Category	JA	–	**61** p90 62 · 115 q · 11.7 MB	**133** p90 140 · 350 q · 16.4 MB
Search	EN	**59** p90 62 · 97 q · 11.2 MB	**62** p90 67 · 108 q · 11.6 MB	**73** p90 78 · 112 q · 12.6 MB
Search	JA	–	**65** p90 69 · 111 q · 11.7 MB	**133** p90 144 · 323 q · 16.7 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	TranslatePress
Home	EN	**85** p90 91 · 63 q · 13.2 MB	**88** p90 90 · 71 q · 13.6 MB	**100** p90 103 · 79 q · 14.7 MB
Home	JA	–	**89** p90 93 · 75 q · 13.7 MB	**142** p90 146 · 200 q · 19.8 MB
Post	EN	**73** p90 77 · 63 q · 13.2 MB	**78** p90 79 · 71 q · 13.6 MB	**86** p90 88 · 82 q · 14.4 MB
Post	JA	–	**77** p90 81 · 72 q · 13.6 MB	**128** p90 135 · 233 q · 16.6 MB
Page	EN	**73** p90 74 · 61 q · 13.2 MB	**77** p90 80 · 70 q · 13.6 MB	**85** p90 86 · 76 q · 14.4 MB
Page	JA	–	**76** p90 78 · 73 q · 13.6 MB	**125** p90 128 · 233 q · 16.6 MB
Category	EN	**79** p90 87 · 76 q · 13.3 MB	**84** p90 90 · 85 q · 13.7 MB	**91** p90 93 · 91 q · 14.5 MB
Category	JA	–	**86** p90 88 · 96 q · 13.7 MB	**134** p90 135 · 256 q · 16.7 MB
Search	EN	**82** p90 84 · 73 q · 13.3 MB	**85** p90 89 · 82 q · 13.7 MB	**93** p90 98 · 89 q · 14.5 MB
Search	JA	–	**86** p90 88 · 84 q · 13.7 MB	**135** p90 140 · 242 q · 16.7 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	TranslatePress
Home	EN	**61** p90 76 · 49 q · 8.9 MB	**65** p90 71 · 57 q · 9.3 MB	**75** p90 79 · 64 q · 10.2 MB
Home	JA	–	**65** p90 69 · 61 q · 9.4 MB	**116** p90 121 · 188 q · 13.9 MB
Post	EN	**60** p90 62 · 60 q · 8.8 MB	**64** p90 68 · 70 q · 9.2 MB	**73** p90 76 · 80 q · 10.2 MB
Post	JA	–	**66** p90 68 · 79 q · 9.3 MB	**113** p90 126 · 216 q · 13.7 MB
Page	EN	**55** p90 58 · 46 q · 8.7 MB	**58** p90 60 · 54 q · 9.1 MB	**67** p90 69 · 66 q · 10.0 MB
Page	JA	–	**59** p90 67 · 57 q · 9.1 MB	**96** p90 103 · 183 q · 12.3 MB
Category	EN	**58** p90 60 · 55 q · 8.8 MB	**63** p90 65 · 65 q · 9.2 MB	**71** p90 72 · 75 q · 10.0 MB
Category	JA	–	**66** p90 68 · 71 q · 9.5 MB	**118** p90 122 · 215 q · 13.9 MB
Search	EN	**70** p90 76 · 57 q · 8.9 MB	**74** p90 77 · 68 q · 9.4 MB	**82** p90 85 · 70 q · 10.3 MB
Search	JA	–	**77** p90 80 · 71 q · 9.6 MB	**146** p90 151 · 235 q · 15.9 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. Mean over the six shop pages, compared with the shop without a multilingual plugin: MULTIGURA +6 ms EN / +7 ms JA, TranslatePress +46 / +107 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, and TranslatePress has to parse all of it.

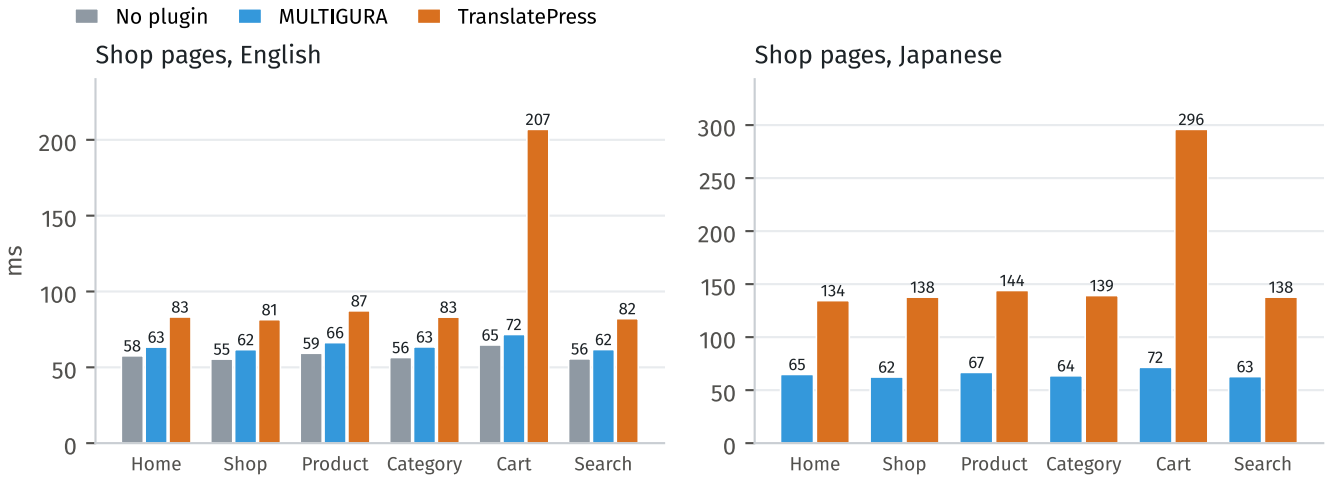

Figure 3. Shop pages, median full-response time in ms (lower is better). Left: English; right: Japanese.

WooCommerce: front-end, ms (median)

Page	Lang	No plugin	MULTIGURA	TranslatePress
Home	EN	**58** p90 58 · 108 q · 11.1 MB	**63** p90 67 · 121 q · 11.7 MB	**83** p90 92 · 121 q · 20.0 MB
Home	JA	–	**65** p90 66 · 131 q · 11.9 MB	**134** p90 142 · 280 q · 24.7 MB
Shop	EN	**55** p90 56 · 80 q · 11.1 MB	**62** p90 64 · 90 q · 11.8 MB	**81** p90 83 · 100 q · 20.1 MB
Shop	JA	–	**62** p90 65 · 94 q · 11.9 MB	**138** p90 140 · 299 q · 24.3 MB
Product	EN	**59** p90 60 · 99 q · 11.1 MB	**66** p90 68 · 119 q · 11.6 MB	**87** p90 90 · 116 q · 20.0 MB
Product	JA	–	**67** p90 69 · 122 q · 11.8 MB	**144** p90 146 · 325 q · 24.3 MB
Category	EN	**56** p90 58 · 84 q · 11.2 MB	**63** p90 66 · 96 q · 11.8 MB	**83** p90 90 · 104 q · 20.1 MB
Category	JA	–	**64** p90 65 · 100 q · 11.9 MB	**139** p90 142 · 309 q · 24.2 MB
Cart	EN	**65** p90 67 · 98 q · 12.1 MB	**72** p90 75 · 108 q · 12.7 MB	**207** p90 211 · 137 q · 26.7 MB
Cart	JA	–	**72** p90 74 · 115 q · 12.8 MB	**296** p90 300 · 328 q · 33.3 MB
Search	EN	**56** p90 57 · 79 q · 11.2 MB	**62** p90 63 · 89 q · 11.7 MB	**82** p90 86 · 96 q · 20.1 MB
Search	JA	–	**63** p90 72 · 94 q · 11.9 MB	**138** p90 140 · 286 q · 24.4 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 37 ms, MULTIGURA 39 ms, TranslatePress 115 ms. Guest add to cart on the Japanese site: MULTIGURA 30 ms, TranslatePress 39 ms (no plugin, English: 25 ms).

Operation	No plugin	MULTIGURA	TranslatePress
Create product	**53** 253 q	**56** 276 q	**132** 276 q
Update product	**29** 52 q	**30** 53 q	**107** 74 q
Save JA product translation	–	**55** 257 q	**34** 89 q
Add to cart (EN, guest)	**25** 50 q	**26** 56 q	**33** 63 q
Add to cart (JA, guest)	–	**30** 61 q	**39** 72 q
Delete product	**29** 52 q	**31** 53 q	**106** 74 q

Shop admin screens (PHP time, ms)

Screen	No plugin	MULTIGURA	TranslatePress
Dashboard	**77** 118 q	**79** 117 q	**89** 150 q
Pages list	**85** 117 q	**93** 120 q	**96** 133 q
Products list	**121** 136 q	**134** 138 q	**131** 152 q
Edit product	**108** 139 q	**114** 154 q	**119** 155 q
Product categories	**98** 116 q	**102** 114 q	**108** 132 q
WooCommerce settings	**92** 111 q	**95** 108 q	**102** 127 q

5. Admin screens

Server time (PHP) of admin screens, mean over the screens of all scenarios: no plugin 89 ms, MULTIGURA 95 ms, TranslatePress 98 ms. Builder editors are dominated by the builder itself. The WooCommerce screens are in section 4.

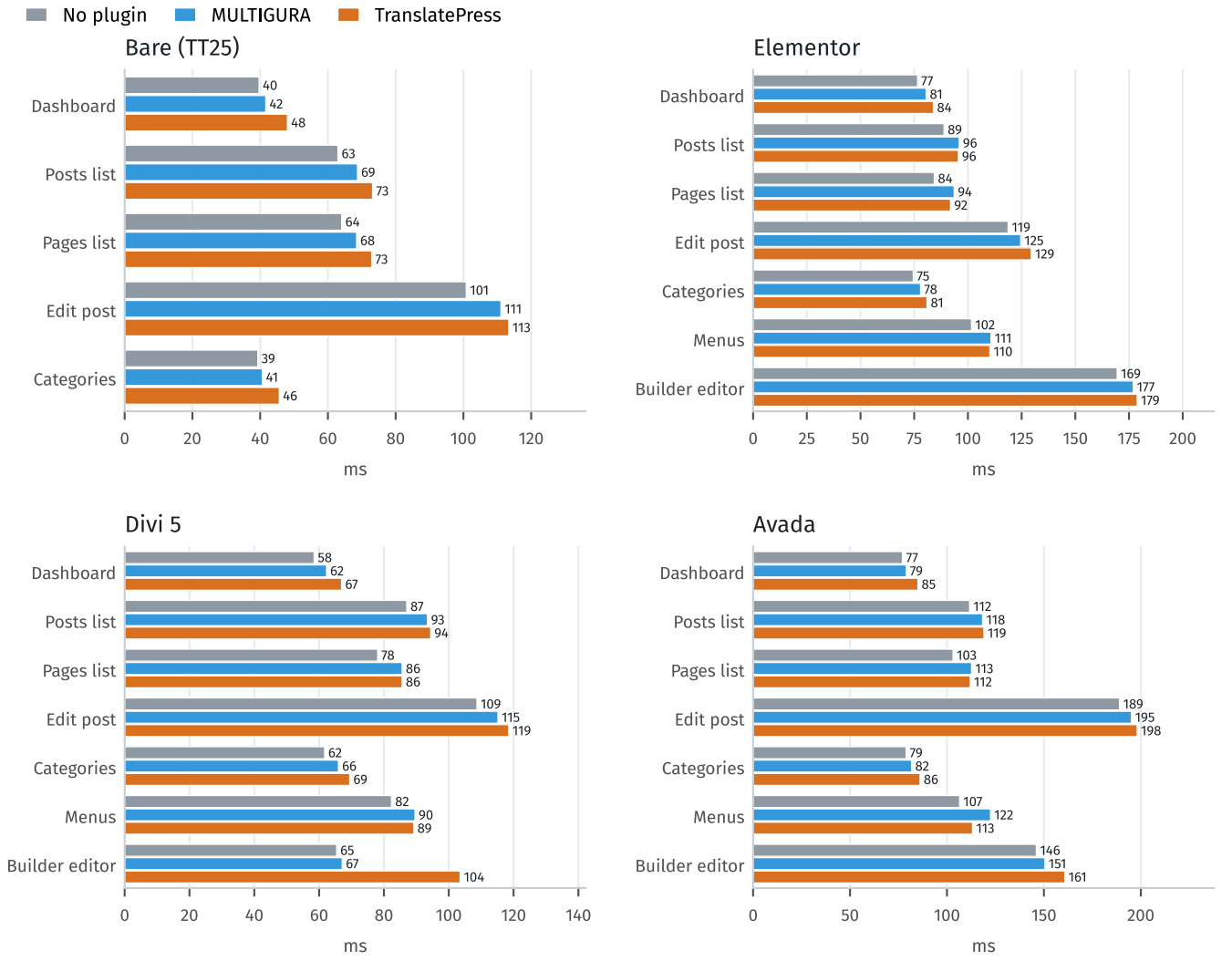

Figure 4. Admin screens, server (PHP) time in ms, median (lower is better).

Scenario	Screen	No plugin	MULTIGURA	TranslatePress
Bare (TT25)	Dashboard	**40** 39 q	**42** 43 q	**48** 71 q
Bare (TT25)	Posts list	**63** 44 q	**69** 48 q	**73** 60 q
Bare (TT25)	Pages list	**64** 36 q	**68** 43 q	**73** 52 q
Bare (TT25)	Edit post	**101** 56 q	**111** 76 q	**113** 72 q
Bare (TT25)	Categories	**39** 29 q	**41** 31 q	**46** 45 q
Elementor	Dashboard	**77** 116 q	**81** 121 q	**84** 148 q
Elementor	Posts list	**89** 115 q	**96** 124 q	**96** 131 q
Elementor	Pages list	**84** 107 q	**94** 136 q	**92** 123 q
Elementor	Edit post	**119** 138 q	**125** 159 q	**129** 154 q
Elementor	Categories	**75** 100 q	**78** 104 q	**81** 116 q

Scenario	Screen	No plugin	MULTIGURA	TranslatePress
Elementor	Menus	**102** 218 q	**111** 226 q	**110** 235 q
Elementor	Builder editor	**169** 140 q	**177** 147 q	**179** 158 q
Divi 5	Dashboard	**58** 55 q	**62** 60 q	**67** 87 q
Divi 5	Posts list	**87** 59 q	**93** 63 q	**94** 75 q
Divi 5	Pages list	**78** 51 q	**86** 57 q	**86** 67 q
Divi 5	Edit post	**109** 83 q	**115** 103 q	**119** 98 q
Divi 5	Categories	**62** 44 q	**66** 48 q	**69** 60 q
Divi 5	Menus	**82** 158 q	**90** 162 q	**89** 175 q
Divi 5	Builder editor	**65** 50 q	**67** 52 q	**104** 70 q
Avada	Dashboard	**77** 71 q	**79** 75 q	**85** 103 q
Avada	Posts list	**112** 74 q	**118** 79 q	**119** 90 q
Avada	Pages list	**103** 66 q	**113** 74 q	**112** 82 q
Avada	Edit post	**189** 129 q	**195** 149 q	**198** 145 q
Avada	Categories	**79** 63 q	**82** 65 q	**86** 79 q
Avada	Menus	**107** 271 q	**122** 364 q	**113** 279 q
Avada	Builder editor	**146** 92 q	**151** 96 q	**161** 104 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 70 ms, MULTIGURA 73 ms, TranslatePress 77 ms. Most of the time is WordPress core (revisions, term counts, caches). The WooCommerce writes are in section 4.

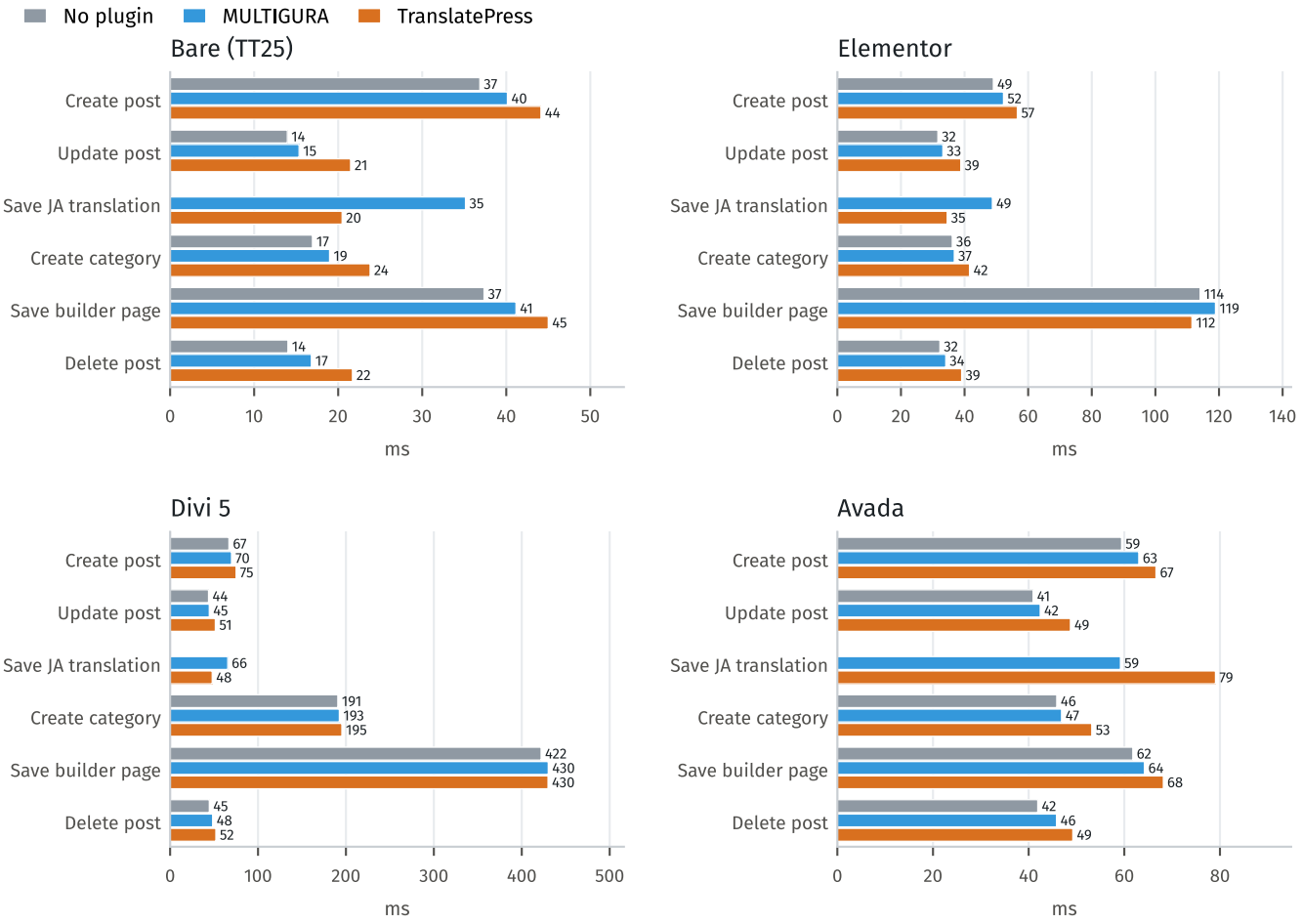

Figure 5. Write requests, server (PHP) time in ms, median (lower is better).

Scenario	Operation	No plugin	MULTIGURA	TranslatePress
Bare (TT25)	Create post	**37** 103 q	**40** 119 q	**44** 114 q
Bare (TT25)	Update post	**14** 10 q	**15** 12 q	**21** 21 q
Bare (TT25)	Save JA translation	–	**35** 69 q	**20** 33 q
Bare (TT25)	Create category	**17** 26 q	**19** 29 q	**24** 38 q
Bare (TT25)	Save builder page	**37** 41 q	**41** 59 q	**45** 52 q
Bare (TT25)	Delete post	**14** 10 q	**17** 17 q	**22** 21 q
Elementor	Create post	**49** 161 q	**52** 179 q	**57** 172 q
Elementor	Update post	**32** 68 q	**33** 70 q	**39** 78 q
Elementor	Save JA translation	–	**49** 129 q	**35** 94 q
Elementor	Create category	**36** 84 q	**37** 87 q	**42** 95 q
Elementor	Save builder page	**114** 269 q	**119** 310 q	**112** 287 q
Elementor	Delete post	**32** 68 q	**34** 77 q	**39** 78 q

Scenario	Operation	No plugin	MULTIGURA	TranslatePress
Divi 5	Create post	**67** 153 q	**70** 173 q	**75** 164 q
Divi 5	Update post	**44** 22 q	**45** 23 q	**51** 32 q
Divi 5	Save JA translation	–	**66** 123 q	**48** 51 q
Divi 5	Create category	**191** 50 q	**193** 53 q	**195** 61 q
Divi 5	Save builder page	**422** 87 q	**430** 107 q	**430** 98 q
Divi 5	Delete post	**45** 22 q	**48** 28 q	**52** 32 q
Avada	Create post	**59** 131 q	**63** 153 q	**67** 142 q
Avada	Update post	**41** 28 q	**42** 30 q	**49** 39 q
Avada	Save JA translation	–	**59** 103 q	**79** 74 q
Avada	Create category	**46** 44 q	**47** 47 q	**53** 56 q
Avada	Save builder page	**62** 69 q	**64** 85 q	**68** 80 q
Avada	Delete post	**42** 28 q	**46** 35 q	**49** 39 q

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

100 posts inserted, updated, translated (MULTIGURA: a JA post through its API; TranslatePress has no translated posts) and deleted in one PHP process; time and queries per item.

Plugin	Insert	Update	Add JA translation	Delete
No plugin	3.1 ms · 37 q	2.4 ms · 23 q	–	3.0 ms · 37 q
MULTIGURA	3.9 ms · 46 q	3.0 ms · 30 q	4.7 ms · 49 q	2.6 ms · 38 q
TranslatePress	3.6 ms · 38 q	3.6 ms · 24 q	–	3.1 ms · 37 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.

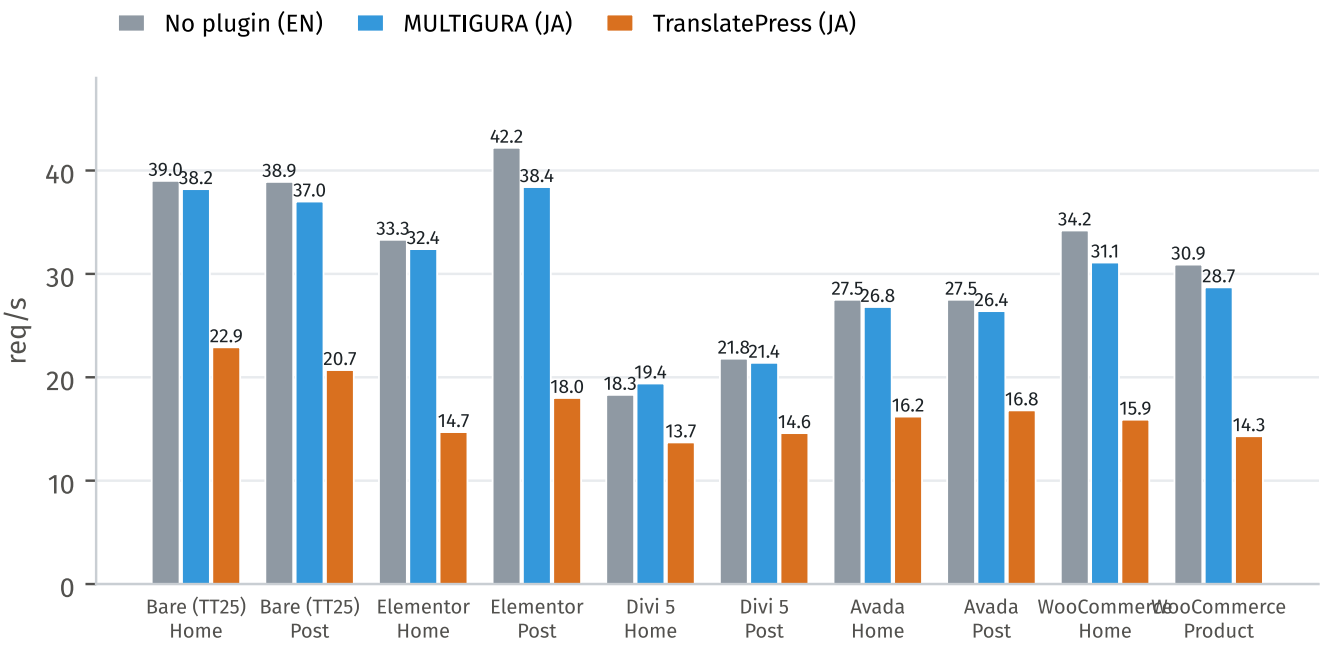

Figure 6. Requests per second with 4 concurrent clients (higher is better).

Scenario	Page	Lang	No plugin	MULTIGURA	TranslatePress
Bare (TT25)	Home	EN	**39.0** med 79 · p95 90 ms	**38.8** med 81 · p95 93 ms	**31.0** med 105 · p95 116 ms
Bare (TT25)	Home	JA	–	**38.2** med 82 · p95 93 ms	**22.9** med 138 · p95 152 ms
Bare (TT25)	Post	EN	**38.9** med 81 · p95 93 ms	**36.8** med 85 · p95 94 ms	**29.9** med 109 · p95 124 ms
Bare (TT25)	Post	JA	–	**37.0** med 85 · p95 97 ms	**20.7** med 155 · p95 168 ms
Elementor	Home	EN	**33.3** med 106 · p95 133 ms	**31.9** med 111 · p95 124 ms	**25.1** med 143 · p95 162 ms
Elementor	Home	JA	–	**32.4** med 112 · p95 124 ms	**14.7** med 226 · p95 240 ms
Elementor	Post	EN	**42.2** med 76 · p95 84 ms	**39.6** med 81 · p95 90 ms	**33.9** med 98 · p95 109 ms
Elementor	Post	JA	–	**38.4** med 83 · p95 94 ms	**18.0** med 180 · p95 196 ms
Divi 5	Home	EN	**18.3** med 155 · p95 169 ms	**18.1** med 163 · p95 176 ms	**16.9** med 179 · p95 202 ms
Divi 5	Home	JA	–	**19.4** med 160 · p95 174 ms	**13.7** med 244 · p95 266 ms
Divi 5	Post	EN	**21.8** med 141 · p95 159 ms	**21.2** med 145 · p95 162 ms	**19.7** med 160 · p95 174 ms
Divi 5	Post	JA	–	**21.4** med 143 · p95 158 ms	**14.6** med 227 · p95 250 ms
Avada	Home	EN	**27.5** med 110 · p95 150 ms	**26.5** med 115 · p95 144 ms	**23.8** med 130 · p95 158 ms
Avada	Home	JA	–	**26.8** med 117 · p95 140 ms	**16.2** med 209 · p95 230 ms
Avada	Post	EN	**27.5** med 114 · p95 141 ms	**26.6** med 118 · p95 143 ms	**24.0** med 134 · p95 156 ms
Avada	Post	JA	–	**26.4** med 118 · p95 144 ms	**16.8** med 198 · p95 224 ms
WooCommerce	Home	EN	**34.2** med 89 · p95 106 ms	**31.8** med 95 · p95 104 ms	**24.6** med 127 · p95 141 ms
WooCommerce	Home	JA	–	**31.1** med 98 · p95 107 ms	**15.9** med 209 · p95 229 ms
WooCommerce	Product	EN	**30.9** med 98 · p95 109 ms	**28.4** med 107 · p95 122 ms	**23.0** med 139 · p95 152 ms

Scenario	Page	Lang	No plugin	MULTIGURA	TranslatePress
WooCommerce	Product	JA	–	**28.7** med 106 · p95 116 ms	**14.3** med 234 · p95 255 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	70	–	10.2	–
MULTIGURA	1 166	1 166	1 149	80	84	10.7	10.7
TranslatePress	1 252	1 255	1 232	88	241	13.6	17.2

9. Findings

- **Overall.** MULTIGURA was faster than TranslatePress in 52 of 52 combinations.
- **Japanese pages:** on average MULTIGURA adds +5 ms and TranslatePress +67 ms over the site without a plugin; TranslatePress needs 1.97× the time of MULTIGURA.
- **Search:** MULTIGURA adds +4 ms to an English search page over the baseline (content scenarios), TranslatePress +14 ms.
- **WooCommerce:** MULTIGURA adds +6 ms (EN) / +7 ms (JA) on shop pages, TranslatePress +46 / +107 ms. The cart: MULTIGURA 72 / 72 ms, TranslatePress 207 / 296 ms (EN / JA).
- **Architecture:** TranslatePress translates after rendering, so its cost grows with the size of the HTML and is paid on every uncached request. MULTIGURA resolves the language before the query runs, so a translated page costs about the same as the original.
- **Code footprint:** 42 extra PHP files per front-end request with MULTIGURA, 128 with TranslatePress.

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. With more languages, TranslatePress keeps a dictionary table per language.
- 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; MULTIGURA's Memcached cache was off.
- The measurement was done by the MULTIGURA team. The scripts and raw results are available on request.