ذاكرة التخزين المؤقت والفهارس موجودة، لكن الصفحة بطيئة: تحقيق في MariaDB داخل PmaControl

Aurélien LEQUOY · 9 أكتوبر 2026

PMACONTROL MARIADB PERFORMANCE SQL BENCHMARK ARCHITECTURE

PmaControl · Enquête performance · Mise à jour du 10 octobre 2026

Le cache était prêt. La recherche était lente.

Un client · 61 graphes · Requête SQL uniquement

Avant · EXISTS

5,04 secondes →

Après · LIMIT 1

3,32 ms

Même contenu : 61 graphes et 995 307 octets de SVG

Instance de test · 30 graphes · Chromium depuis un serveur sur le réseau local

Avant · Chargement complet

523,15 ms →

Après · Chargement complet

488,40 ms

Téléchargement du document : 42,90 → 37,45 ms · SVG identiques

Médianes de 10 passages par variante. Deux campagnes distinctes ; temps non additionnables.
Sur l’instance de test, le plan initial était déjà rapide. Les séries navigateur se recouvrent.

«الرسومات موجودة بالفعل في ذاكرة التخزين المؤقت. لماذا لا يزال عرض صفحة البنية يستغرق عدة ثوانٍ؟»

قادنا هذا السؤال إلى إصلاح قصير جدًا في PmaControl. **لم يكن هناك فهرس مفقود، ولم تكن هناك حاجة إلى إعادة حساب الرسومات.** اختارت MariaDB طريقة مكلفة للعثور على جيل الرسوم البيانية المطلوب عرضه.

فيما يلي المقاييس وSQL والقيود المفروضة على هذه المقارنة. نحن نميز بين **العميل** الذي قدم المشكلة ومثيل الاختبار الذي كان ينفذ الاستعلام الأولي بسرعة بالفعل.

تحديث 10 أكتوبر 2026. أضيفت أداة قياسات SQL المسجلة عند الساعة 1:29 مساءً وأعيد قياس أوقات التنزيل قرابة الساعة 1:45 مساءً من خادم داخل الشبكة المحلية، لأن الاتصال المستخدم للالتقاط قياسات 9 أكتوبر كان غير مستقر. تحتفظ كل حملة ببيئتها وعدد

رسوماه ونطاقاه: SQL أو تنزيل HTTP أو التحميل في Chromium.

شرح المشكلة بصندوق من الرسومات

تخيل مكتبة مليئة بالصناديق. يحتوي كل صندوق على رسومات لجيل واحد من الطوبولوجيا. تم الانتهاء من الرسومات بالفعل: هذه هي ذاكرة التخزين المؤقت لـ SVG الخاصة بنا.

لعرض الصفحة، يبقى على التطبيق العثور على الصندوق الصحيح وأخذ رسوماته ثم ترتيبها على الشاشة. فهرس المكتبة موجود بالفعل، وهو فهرس SQL.

ولكن وجود كتالوج ليس كافيا. في الحالة البطيئة، اختارت MariaDB تكرار ملايين الإدخالات لإعداد قائمة وسيطة من التطابقات. بالنسبة للتحقق الذي يهمنا، يكفي العثور على **تطابق واحد**.

التصحيح يجعل هذا الاختيار واضحا. يتم إرجاع نفس الرسومات؛ البحث يتجنب الانعطاف المكلف.

ما تم تخزينه مؤقتاً بالفعل

يتم حفظ ملفات SVG في `dot3_graph`. يربطهم الجدول `dot3_cluster` بأجيال الطوبولوجيا. تسترد الصفحة بالفعل جميع الرسوم البيانية باستخدام **استعلام SQL واحد**، والذي يُرجع بشكل خاص:

```
SELECT dg.id, dg.svg, dg.height, dg.width,
       (dg.height * dg.width) AS area,
       dc.date_inserted AS date_refresh
FROM dot3_cluster dc
JOIN dot3_graph dg ON dg.id = dc.id_dot3_graph
-- يختار الشرط الجيل الجاهز للعرض.
```

وبالتالي فإن المتصفح يستقبل الرسم وارتفاعه وعرضه وسطحه. يقوم رمز `App/Webroot/js/Architecture/topology.js` بالفعل بوضعه في المتصفح. هذا التصحيح لا يعدل هذه الخوارزمية.

تمنع ذاكرة التخزين المؤقت إعادة إنشاء الرسومات في كل زيارة، لكنها لا تمنع البحث عنها ونقلها وعرضها. ذاكرة SVG المؤقتة هذه منفصلة عن ذاكرة استعلامات MariaDB المؤقتة وعن ذاكرة HTTP المؤقتة في المتصفح.

لا، لم يكن هناك أي فهرس مفقود

يحتوي المثيل البطيء بالفعل على هذا الفهرس الفريد، كما تم تأكيده بواسطة `SHOW INDEX FROM dot3_cluster`:

```
-- الفهرس الموجود، معروضًا في صورة تعريفه:
UNIQUE KEY id_dot3_information_2
    (id_dot3_information, id_dot3_graph)
```

مرشحات البحث على `id_dot3_information`، العمود الأول من الفهرس. وهذا يجعل من الممكن العثور على مجموعات من الجيل.

كانت المشكلة في **خطة التنفيذ**، أي الاستراتيجية التي اختارها MariaDB لتنفيذ SQL. استخدم الإصدار الأول `EXISTS` للتحقق من أن الجيل يحتوي على مجموعة واحدة على الأقل. في هذا المثيل أدى الاختيار إلى عملية materialization: أعدت MariaDB نتيجة وسيطة بعد مسح نحو **4.33 مليون إدخال تاريخي**.

لا يعني ذلك أن المتصفح تلقى أربعة ملايين رسم بياني؛ فقد تلقى **61 رسمًا بيانيًا** فقط. حدث العمل الزائد داخل قاعدة البيانات قبل استرجاع النتيجة المفيدة. يصف رقم 4.33 مليون حجم المسح المكتشف في التشخيص الأول، وليس عدد الصفوف المنقولة إلى PHP. وتُظهر قياسات 10 أكتوبر نحو **5.2 مليون صف** في جدول هذا المثيل؛ لذلك لا يتطابق حجم بيانات المقالة الأولى مع الحملة الجديدة تمامًا.

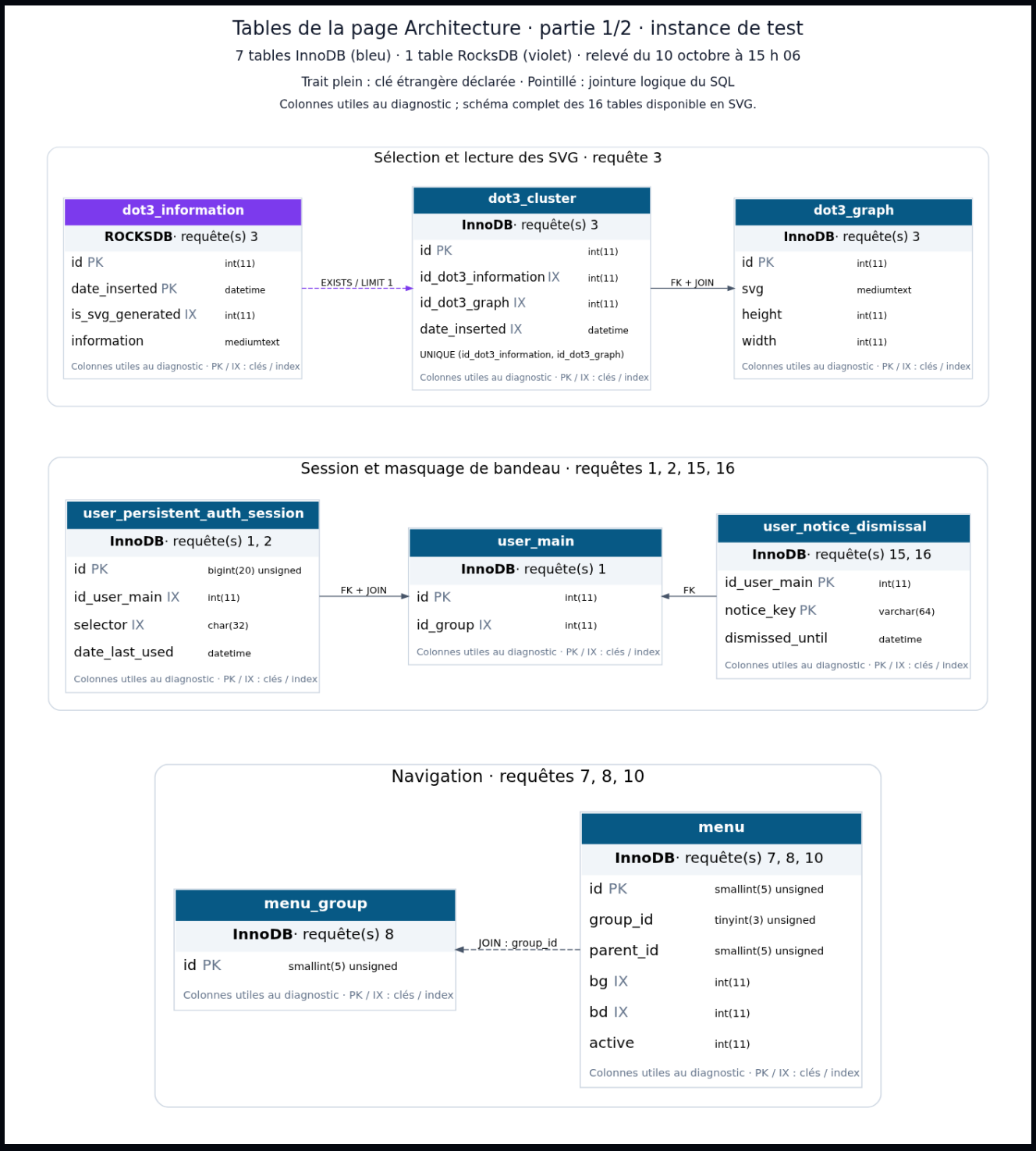
يمكن استخدام الفهرس أثناء إجراء فحص واسع جدًا. وبالتالي فإن وجود الفهرس لا يضمن في حد ذاته الوصول القصير.

تكوين وجداول MariaDB وInnoDB / RocksDB

تستخدم كلا البيئتين **MariaDB 11.8.9**، مع حزم مختلفة: **11.8.9-deb13 على العميل**، وقفًا لتقريرها الصادر في 10 أكتوبر، و**11.8.9-MariaDB-deb12-log على نسخة الاختبار**، والتي تم التحقق منها مباشرة على الخادم. يستخدم العميل PHP 8.4 لإعادة تشغيل SQL؛ يستخدم مثيل الاختبار FrankenPHP / PHP 8.5.9.

يغطي الرسم البياني أدناه **جداول التطبيق الستة عشر التي تستخدمها الاستعلامات الستة عشر الموجودة على الصفحة**، بالإضافة إلى الجداول الثلاثة في استعلام SVG. تأتي الأعمدة والمفاتيح والمحركات المفيدة من الرسم التخطيطي الفعلي لمثيل الاختبار، المذكور في **10 أكتوبر الساعة 3:06 مساءً**. تقوم عمليات التحقق من وجود الجدول أيضًا بالاستعلام عن `information_schema.tables`، وهو عرض بيانات تعريف ليس جدول

تطبيق لتصنيفه على أنه InnoDB أو RocksDB.





مخطط متجه مفتوح كبير. يشير الخط المتصل إلى مفتاح خارجي مُعلن؛ يشير الخط المنقط إلى صلة منطقية تستخدمها SQL. يتم الاستعلام عن الجداول التي لا تحتوي على ارتباط معروض بشكل منفصل. يوضح الرسم العلاقات، وليس ترتيب تنفيذ الاستعلام.

جدول استعلام SVG في مثيل الاختبار	محرك التخزين المرصود
`dot3_information`	**RocksDB**
`dot3_cluster`	**InnoDB**
`dot3_graph`	**InnoDB**

تستخدم الجداول الثلاثة عشر الأخرى في الصفحة محرك InnoDB. ويعرض المخطط الكامل أعلاه و جرد محركات التخزين والأحجام القابل للتنزيل تفاصيلها.

ولذلك يمر طلب SVG عبر **محركين** في نسخة الاختبار. ينتمي مؤشر توفر `dot3_information` إلى RocksDB؛ ينتمي الفهرس الفريد لـ `dot3_cluster` وSVGs لـ `dot3_graph` إلى InnoDB. الإصلاح يحافظ على هذه المحركات والفهارس. تقارن هذه الحملة بين صيغتين SQL؛ والقياس تحويل الجداول من InnoDB إلى RocksDB. القدم بيانات العميل محركاته: نحن لا ننسب إليهم النمط الذي لوحظ في الاختبار.

يجمع ملف إعداد `.cnf` المجهول لمثيل الاختبار أقسام الخادم من الملفات المضمّنة حسب ترتيب تحميلها، بعد إزالة الأسرار ومعرّفات الشبكة. وبما أن الإعدادات اللاحقة قد تتجاوز السابقة، تعرض المتغيرات العامة المقاسة فعلياً القيم النشطة، ومنها **2 GiB لمخزن InnoDB المؤقت**، و**2 GiB ذاكرة كتل RocksDB المؤقتة**، و**128 اتصالاً كحد أقصى**، و**64 MiB لكل من `tmp_table_size` و `max_heap_table_size`**، مع **تعطيل ذاكرة الاستعلامات المؤقتة**.

جُمعت هذه المعلمات عند الساعة **3:14 مساءً**، بعد قياسات الساعة 1:45؛ وقد تتغير بعض المتغيرات ديناميكياً. يصف مورد `.cnf` الإعداد المرصود وتجاوزاته بعد حذف القيم الحساسة، وليس ملفاً يُطبّق مباشرة. لم تتضمن القياسات إعداد `.cnf` الخاص بالعميل.

تعديل SQL، قبل وبعد

يقع التغيير في `App/Library/Dot3LatestGeneration.php`، ضمن اختيار الجيل الأحدث الجاهز.

قبل :

```
AND EXISTS (
    SELECT 1
    FROM dot3_cluster dc_ready
    WHERE dc_ready.id_dot3_information = di.is_svg_generated
)
```

بعد :

```
AND (
    SELECT dc_ready.id_dot3_information
    FROM dot3_cluster dc_ready
    WHERE dc_ready.id_dot3_information = di.is_svg_generated
    LIMIT 1
```

```
) IS NOT NULL
```

لماذا هما متكافئان هنا؟ يفرض الاستعلام الخارجي أصلاً `di.is_svg_generated > 0`. إذا وُجدت مجموعة مطابقة يعيد الاستعلام الفرعي هذا المعرّف الموجب، وإلا يعيد الاستعلام الفرعي القياسي بلا نتائج القيمة `NULL`. لذلك يظل الشرط يتحقق من وجود مجموعة لهذا الجيل.

يتجنب الاستعلام الفرعي القياسي خطوة materialization التي لوحظت مع `EXISTS`. يستطيع المحرك البحث عن الجيل في الفهرس والتوقف عند أول تطابق. وتبقى مسارات الرجوع المستخدمة عند غياب جيل جاهز كما هي.

هذه ليست قاعدة تقول أن `EXISTS` بطيء. وفي المثال الآخر، كان الاستعلام القديم يستخدم بالفعل إستراتيجية `FirstMatch` الفعالة. إعادة كتابة كل `EXISTS` ستكون نتيجة سيئة: عليك فحص وقياس خطة كل استعلام.

عميل: قياسات SQL جديدة في 10 أكتوبر

تقارن القياسات المنفذة لدى **عميل** قرابة **1:29 مساءً** عشر مرات تشغيل لكل صيغة على بيانات الإنتاج، بعد استبعاد تشغيل إحماء واحد والتناوب بين الترتيبين قبل/بعد. يتضمن كود الإنتاج الإصلاح بالفعل؛ تعني «قبل» هنا تشغيل استعلام `EXISTS` القديم للمقارنة، وتعني «بعد» الاستعلام الفرعي القياسي مع `LIMIT 1`.

قياس SQL	قبل (`EXISTS`)	بعد (`LIMIT 1`)
استعلام SVG، الحد الأدنى – الحد الأقصى	4 434,17–5 297,38 ms	2,69–6,20 ms
استعلام SVG، متوسط	**5 044,59 ms**	**3,32 ms**
إجمالي 16 طلباً، متوسط	5 068,52 ms	12,92 ms
الرسوم البيانية التي تم إرجاعها	61	61
حجم SVG	995 307 بايت	995 307 بايت
عداد قراءة فهرس البيان	5 199 176	3

ينخفض زمن الاستعلام المهيمن من **5.04 ثانية إلى 3.32 ملي ثانية**، أي يصبح أسرع بنحو **1,520 مرة** في هذه القياسات. الرسوم البيانية الـ61 متطابقة قبل الإصلاح وبعده. ويختلف الحجم الحالي بمقدار 130 بايت عن التشخيص الأول الذي سجل 995,437 بايت؛ فهما تاريخا قياس مختلفان، بينما تعيد صيغتا الحمولة الجديدة المحتوى نفسه.

تُقدّر خطة `EXPLAIN` قبل الإصلاح **4,406,092 إدخالاً** لعملية materialization، بينما تُقدّر بعد الإصلاح **81 إدخالاً** للاستعلام الفرعي التابع الذي يستخدم الفهرس. هذه التقديرات ليست عدد القراءات المنفذة فعلياً. يسجل عداد قراءات الفهرس **5,199,176 قبل الإصلاح و3 بعده**؛ وكان `dot3_cluster` يحتوي حينها على **5,198,990 صفاً**. تقدير الخطة والقراءات المرصودة والرسوم المنقولة ثلاثة مقاييس مختلفة.

متوسطات الطلبات الـ 15 الأخرى تظل جميعها أقل من 10 ملي ثانية. يتقلب بعضها: يتراوح عدد الإنذارات من 9.21 إلى 0.85 ملي ثانية، على سبيل المثال، ويصل أحياناً إلى 15.55 ملي ثانية قبل ذلك. لا توضح هذه النتيجة أن الإصلاح يسرع عملية العد: كان لدى الخادم تحميل يبلغ حوالي 6 وكانت هذه الطلبات تتبع مباشرة طلباً مدته خمس ثوانٍ يتعلق تغيير الكود فقط باختيار ملفات SVG.

نُفذ استعلام التعديل الوحيد في هذه الحملة، وهو تحديث الجلسة، ثم أُلغي. **يقيس هذا التقرير SQL؛ ولا يقيس التنزيل أو التحميل الكامل في المتصفح.** تشمل الأوقات استرجاع mysqli الكامل للنتائج، ومنها نحو 1 ميغابايت من SVG.

طلب من العميل	قبل (ملي ثانية)	بعد (ملي ثانية)
1. اقرأ جلسة تسجيل الدخول	0,34	0,34
2. قم بتحديث الجلسة، ثم قم بإلغاء	0,16	0,19
3. قم بتحميل الرسوم البيانية الـ 61 وأبعادها	5 044,59	3,32
4. قائمة العملاء	0,53	0,29
5. قائمة البيئات	0,78	0,15
6. قائمة العلامات	0,60	0,22
7. ابحث عن موجز التنقل	1,59	0,64
8. قم ببناء القائمة	5,26	4,24
9. عد الإنذارات النشطة	9,21	0,85
10. حدد عنصر القائمة	0,78	0,66
11. تحقق من مفاتيح الترجمة	0,53	0,15
12. اقرأ الترجمة	0,48	0,18
13. تحقق من جدول التكوين	0,20	0,17

طلب من العميل	قبل (مللي ثانية)	بعد (مللي ثانية)
14. اقرأ إعدادات الترخيص	0,17	0,15
15. تحقق من جدول الشعارات المخفية	0,18	0,15
16. اقرأ إخفاء شعار الترخيص	0,30	0,13
الإجمالي، وسيط مجموع كل تشغيل	**5 068,52**	**12,92**

قياسات SQL مجهولة الهوية مع نطاقات الزمن وخطتي التنفيذ تكمل هذا الجدول.

زمن التنزيل لدى العميل: فرضية للمقارنة

لإكمال مقارنة HTTP، **نأخذ أوقات التنزيل التي تم قياسها على نسخة الاختبار من الخادم**، والموضحة في القسم التالي. نحن نفترض أن هذه الفترات متطابقة بالنسبة للعميل. **هذه القيمة هي افتراض مشترك بين البيئتين؛ فهي لا تشكل مقياس HTTP جديداً لدى العميل.**

تنزيل مستند HTML، معاد استخدامه كفرضية	قبل	بعد
وسيط Chromium في مثيل الاختبار	42,90 ms	37,45 ms

لم يُعَدَّ قياس زمن البايت الأول أو التحميل الكامل لدى العميل. ولا يمكن استخدام إجماليات الاختبار لأن استعلام SVG وحده كان يستغرق خمس ثوانٍ قبل الإصلاح لدى العميل. لذلك نفصل SQL المقاس عن النقل المفترض ولا نختلق زمناً للمتصفح.

التنزيل والمتصفح: قياس جديد من الخادم

لإزالة الشك الناتج عن اتصال اللقطات القديمة، نُفذت حملة أخرى في **10 أكتوبر قرابة الساعة 1:45 ظهراً** من خادم Debian 12 إلى مثيل الاختبار على الشبكة المحلية. استخدمت صفحة البنية نفسها بلا مرشح، والحساب المؤقت نفسه للقراءة فقط، و**30 رسماً بيانياً**، ونافذة 1440 × 900 بكسل، وChromium 152. عُطلت ذاكرة HTTP المؤقتة في المتصفح، واستُبعد تحميل إحماء واحد لكل صيغة، ثم حُفظت **عشر عمليات تحميل قبل الإصلاح وعشر بعده**.

يستخدم الخادم الهدف MariaDB 11.8.9 وFrankenPHP / PHP 8.5.9. كود التطبيق هو الالتزام `4207dfbe5e7d1709dbbd2efdd2cff8c1e6619dc1`؛ وفي صيغة «قبل» فقط يأتي `Dot3LatestGeneration.php` من الالتزام الأب `366b200703ff7550ec01d33e2c7d49c0b570ab15`. تمر القياسات عبر خادم HTTPS مؤقت يعرض شجرة العمل على الشبكة المحلية. ويظل الإنتاج على

الكود المصحح.

المتوسطات بالمللي ثانية:

المرحلة المقاسة في Chromium	قبل (EXISTS)	بعد (LIMIT 1)
البايت الأول، PHP والنقل متضمن	62,35	50,90
تحميل وثيقة HTML	42,90	37,45
تم استلام الوثيقة → الحدث `load`	416,05	392,20
تحميل كامل → `load`	**523,15**	**488,40**



يختلف تنزيل المستند من **33.2 إلى 61.3 مللي ثانية قبل**، ومن **32.3 إلى 55.1 مللي ثانية بعد**. يختلف التحميل الكامل من **472.3 إلى 681.5 مللي ثانية قبل**، ومن **444.5 إلى 669.7 مللي ثانية بعد**. تتداخل هذه السلاسل: فهي تُظهر تحميلاً يبلغ حوالي نصف ثانية في هذه الرحلة، دون إنشاء كسب متصفح قابل للتكرار يبلغ 34.75 مللي ثانية.

تحمل **ملفات SVG الثلاثون المعروضة قيمة SHA-256 نفسها في كل عملية من عمليات التحميل العشرين**. يبلغ حجم المستند بعد فك الضغط **2,240,909 بايت** في الصيغتين. ويبلغ جسمه المضغوط نحو **671,254 بايت**، أو نحو 671,554 بايت مع الترويسات التي يحسبها Chromium. يخص حجم HTML هذا والرسوم الثلاثون مثيل الاختبار، والي يطابق 995,307 بايت من SVG للرسوم الـ61 لدى العميل.

يؤكد فحص منفصل باستخدام `curl --compressed`، عشرة طلبات لكل متغير بعد التسخين، أن النقل يظل متغيراً: متوسط التنزيل **70.85 مللي ثانية قبل و93.80 مللي ثانية بعد**، واستجابة HTTP الكاملة **119.97 و145.62 مللي ثانية**. الـ `curl` يتحمل مواد الصفحة وتحميل الخادم في حدود تناقضات مع Chromium؛ لا تثبت هذه الاختلافات انحدار SQL.

تم قياس المعالم لكل زيارة ثم تم تلخيصها بشكل منفصل: **مجموع المتوسطات ليس بالضرورة متوسط الإجمالي**. يتضمن الوقت بعد استلام المستند مواد أخرى، وتحليل HTML، وJavaScript، والعرض. نحن لا نعزو ذلك إلى وضع الرسوم البيانية وحده. يتضمن البايت الأول بالفعل إعداد PHP؛ لا تتم إضافة وقت PHP المقدر إلى الإجمالي.

تبقى أزمنة التحميل القديمة 12,948 و2,952 و843 مللي ثانية ملاحظات تاريخية من 9 أكتوبر، وهي متاحة في ملف CSV المؤرشف. اعتمد زمن تنزيلها بين 470 و535 مللي ثانية على الاتصال غير المستقر آنذاك، لذلك لم يعد مرجعاً للشبكة. ولا تعيد الحملة الجديدة ذات 30 رسماً بناء زمن المتصفح للمثيل ذي 61 رسماً.

مقارنة SQL الجديدة في مثيل الاختبار، عشرة أزواج متوازية في قطعة واحدة للقراءة فقط مع تعطيل ذاكرة التخزين المؤقت لاستعلام MariaDB للجلسة، تؤدي إلى **6.94 مللي ثانية بعد و5.81 مللي ثانية قبل**. النتائج الكاملة و**2,172,396 بايت من SVG** متطابقة. لا تزال الخطة الأولية هناك تستخدم `FirstMatch`: حتى مع وجود 6,591,509 سطراً في `dot3_cluster`، فإن هذا المثيل لا يعيد إنتاج التجسيد الباهظ الثمن للخادم الآخر. وبالتالي فإن حجم الطاولة وحده لا يكفي للتنبؤ بالبطء.

مثيل الاختبار: تفاصيل الاستعلامات الـ16 في 9 أكتوبر

يحتفظ الجدول التفصيلي أدناه بالحملة المُجهزة بتاريخ **9 أكتوبر**، على الصفحة بدون فلتر، مع عدد للقراءة فقط و**30 رسماً بيانياً**. تم استبعاد حمولة التسخين الأولى، ثم تم الاحتفاظ بخمسة أحمال لكل إصدار. لم تتم تقديم هذا الجدول كمقياس جديد للطلبات الستة عشر المقدمة في 10 أكتوبر.

يكتشف ملف تعريف التطبيق **16 طلباً في كل عملية تحميل من عشرة**. يتوافق كل سطر أدناه مع استعلام تم تنفيذه مرة واحدة قبله ومرة بعده؛ السطر الأخير يظهر المجموع. الأوقات أدناه هي متوسطات استدعاءات SQL التي تم قياسها على جانب برنامج تشغيل PHP، بالمللي ثانية. إنها لا تمثل فقط الوقت الداخلي لمحرك MariaDB.

استعلام	قبل (مللي ثانية)	بعد (مللي ثانية)
1. اقرأ جلسة تسجيل الدخول	0,31	0,34
2. جلسة التحديث	0,29	0,27
3. قم بتحميل الرسوم البيانية الثلاثين وأبعادها	5,86	5,57
4. قائمة العملاء	0,37	0,34
5. قائمة البيئات	0,23	0,27
6. قائمة العلامات	0,23	0,29
7. ابحث عن موجز التنقل	0,71	1,07
8. قم ببناء القائمة	3,02	3,04
9. عد الإنذارات النشطة	1,11	1,09
10. حدد عنصر القائمة	0,67	0,67
11. تحقق من مفتاح الترجمة	0,33	0,31
12. اقرأ الترجمة	0,30	0,29
13. تحقق من جدول التكوين	0,33	0,31
14. اقرأ إعدادات الترخيص	0,22	0,23
15. تحقق من جدول الشعارات المخفية	0,32	0,31
16. اقرأ إخفاء شعار الترخيص	0,18	0,22
الإجمالي، المتوسط	**14,53**	**14,59**

الإجمالي هو متوسط المبالغ المحسوبة لكل تحميل. إن إضافة متوسطات كل صف لا يعطي بالضرورة هذه القيمة.

في هذا الاختبار، **لم يتم توضيح أي مكاسب إجمالية كبيرة لـ SQL**. الفرق 0.06 مللي ثانية هو الحد الأدنى. تم توضيح ذلك: اختارت MariaDB بالفعل خطة جيدة للاستعلام القديم. يستهدف الإصلاح الخطة المكلفة التي تمت مواجهتها في مثيل الإنتاج.

يغطي هذا العدد طلبات التطبيق الخاصة بعرض PHP الأولي، على الاتصالات الثلاثة التي لاحظها منشئ ملفات التعريف. وهو يستثني مكالمات AJAX اللاحقة وأوامر تسجيل الدخول

الدالية. ال ينبغي تعيين هذه الطلبات الستة عشر لجميع ملفات تعريف المستخدمين أو كافة عمليات التثبيت.

تأكد أيضاً من أن النتائج تظل صحيحة

تكون الصفحة الأسرع مفيدة فقط إذا كانت تعرض الرسوم البيانية الصحيحة. ركز التحقق على المساواة في النتائج قبل/بعد، والأجيال الجاهزة أو الغائبة، والحلول الاحتياطية.

تم تنفيذ تسعة سيناريوهات باستخدام بيانات الاختبار المحددة في MariaDB CTEs، دون إدراجها في جداول التطبيق. غطت مقارنات SQL أيضاً الصفحة التي لم تتم تصفيتها، والصفحة التي تمت تصفيتها، وتجميع الرسم البياني، مع مقارنة النتائج في نفس لقطة المعاملات. لقد تحقق اختبار JavaScript للموضع من السلوك عبر عروض شاشات متعددة.

أشارت سلسلة منفصلة مكونة من خمسة عشر قياساً مقترناً في نسخة الاختبار إلى حملات متوسطة صغيرة على المسارات المصفاة والمجمعة: حوالي **0.34 مللي ثانية** و**0.16 مللي ثانية**. ونحن نحتفظ بها في الميزانية العمومية: فالتصحيح الذي يجعل كل السيناريوهات أسرع. الـ تحل عمليات الفحص المستهدفة هذه محل مجموعة اختبار المنتج بالكامل.

أعد التشخيص في صفحة أخرى

ابدأ بفصل ثلاثة أسئلة: كم من الوقت يقوم الخادم بإعداد الاستجابة، وكم من الوقت يستغرق وصول المستند، وكم من الوقت يستغرق المتصفح لإنهاء التحميل؟

ثم قم بإحصاء الطلبات وقياس كل منها. بالنسبة للشخص الذي يهيمن، قارن بين الخطة السابقة/البعدية والمساواة في النتائج. تحقق بشكل خاص من عدد الإدخالات التي تم البحث عنها، والنتائج المتوسطة والاستخدام الفعال للفهارس. كرر القياسات: ملاحظة واحدة يمكن أن تكون مضللة.

في هذه الحالة، **يظل الاستعلام بمثابة استعلام**، ويظل الفهرس كما هو. لقد تغيرت طريقة البحث عن الرسومات الموجودة بالفعل في ذاكرة التخزين المؤقت.

البيانات التفصيلية وSQL

- الاستعلامات الـ 16 وموقعها في الكود وSQL الكاملة قبل/بعد
- الأحمال العشرة: مدة كل طلب في ملف CSV
- عمليات تحميل Chromium العشرين بتاريخ 10 أكتوبر من الخادم، بتنسيق CSV
- تحكم تجعيد HTTP في 10 أكتوبر، قبل وبعد

- مقارنة SQL بتاريخ 10 أكتوبر على نسخة الاختبار: النتائج والخطط في JSON
- بيان SQL مجهول المصدر من العميل: 16 استعلامًا ونطاقات وخطة
- تكوين MariaDB `.cnf` المجهول لمثيل الاختبار والمتغيرات النشطة والإصدار والملفات المضمنة
- مخطط الجداول الـ 16 في SVG، تعريفات SQL، جرد المحركات والأحجام في CSV والأعمدة والفهارس والعلاقات في JSON
- برنامج نصي SQL، متوفر في النص، برنامج Chromium النصي، برنامج نصي HTTP وتكوين خادم HTTPS المؤقت
- أسلوب وخطوات استنساخ حمله 10 أكتوبر
- الأرشيف: أحمال الإنتاج الثلاثة بتاريخ 9 أكتوبر على الاتصال غير المستقر

يتم استبدال معرفات الجلسة والحساب بمعلمات في الأمثلة القابلة للتنزيل. يتم فصل قياسات الإنتاج والاختبار للسماح بتفسيرها دون الخلط بين البيئتين.