قراءة واحدة من MariaDB للتعرف على ملاحظة تنبيه مسجّلة سابقاً

Aurélien LEQUOY · 21 نوفمبر 2026

PMACONTROL MARIADB MYROCKS PERFORMANCE SQL BENCHMARK

عندما يتلقى PmaControl القياس نفسه مرتين، عليه التعرف على التكرار حتى لا يفتح الحادث أو يغلقه مرة أخرى. كان التطبيق يسأل MariaDB أولاً عن تاريخ أحدث قياس، ثم يبحث عن النقطة التي وصلت للتو. **حذفنا القراءة الأولى.**

في خمسة فحوص حقيقية انخفض وسيط زمن قراءات SQL عند إعادة المعالجة **من 6.645 إلى 0.363 ملي ثانية، أي بنسبة 95%**. وفي المختبر انخفض وسيط المعاملة كاملة، مع القفل والكتابة، **من 2.554 إلى 0.904 ملي ثانية**. اختُبر تعديل التطبيق وهو بانتظار النشر.

فحص المناطق الزمنية خطوة بخطوة

في 21 نوفمبر عند 22:29:19 أشار فحص توافق المناطق الزمنية إلى إنذار. ماذا يحدث إن وصل القياس نفسه مجدداً؟ تبدأ النسخة الأصلية بالاستعلام التالي. تمثل `?` بصمة الفحص، ولها القيمة نفسها في الاستعلامين.

```
EXPLAIN PARTITIONS
SELECT observed_at
FROM pmc_alert_level_point
WHERE fingerprint = UNHEX(?)
ORDER BY observed_epoch DESC, observed_at DESC
LIMIT 1;
```

كان التطبيق يكتفي بوجود صف ليعرف أن للفحص سجلاً، ثم يبحث عن القياس المطابق ويقرأ حالته. تبدأ النسخة المختارة من هذه الخطوة. تطابق النقطة والحالة يعني إعادة معالجة، أما تعارض حالتين في اللحظة نفسها فيؤدي إلى خطأ.

```
EXPLAIN PARTITIONS
SELECT is_open
FROM pmc_alert_level_point
WHERE fingerprint = UNHEX(?)
  AND observed_epoch = 1795296559
```

```
  AND observed_at = '2026-11-21 22:29:19.000000';
```

خطتا التنفيذ من الخادم للفحص نفسه. يعرض عمود النتائج الأول القراءة المحذوفة، والثاني القراءة التي تكفي للتعرف على التكرار.

EXPLAIN PARTITIONS	أحدث تاريخ	النقطة المطابقة
الوصول (`type`)	`range`	`const`
الفهرس (`key`)	`PRIMARY`	`PRIMARY`
بايتات المفتاح (`key_len`)	32	48
المرجع (`ref`)	—	`const,const,const`
الصفوف المقدرة (`rows`)	8,617	1
الأقسام في الخطة	45	1 (`p740307`)
Extra	`Using where; Using index`	—

يستخدم الاستعلامان `PRIMARY`. يحدد الأول البصمة ذات 32 بايت فقط وتظهر الأقسام الـ45 في خطته. يحدد الثاني أيضاً قيمة epoch والتاريخ فيشمل 48 بايت، أي المفتاح بكامله. تستطيع MariaDB تحديد مفتاح واحد في قسم يوم القياس.

القيمة 8,617 في `rows` تقدير. عند تنفيذ `ANALYZE FORMAT=JSON` أعاد مكرّر الاستعلام الأول صفاً واحداً (`r_rows = 1` و `r_loops = 1`) كما يقتضي `LIMIT 1`. لا تقيس هذه العدادات عمليات البحث الداخلية في الأقسام. بين 19 و21 نوفمبر سجّل الملخص **1,643,071 استدعاء** بزمن SQL تراكمي بلغ **8 ساعات و8 دقائق**.

فتح خطط MariaDB العشرين للفحوص الخمسة (ملف نصي)

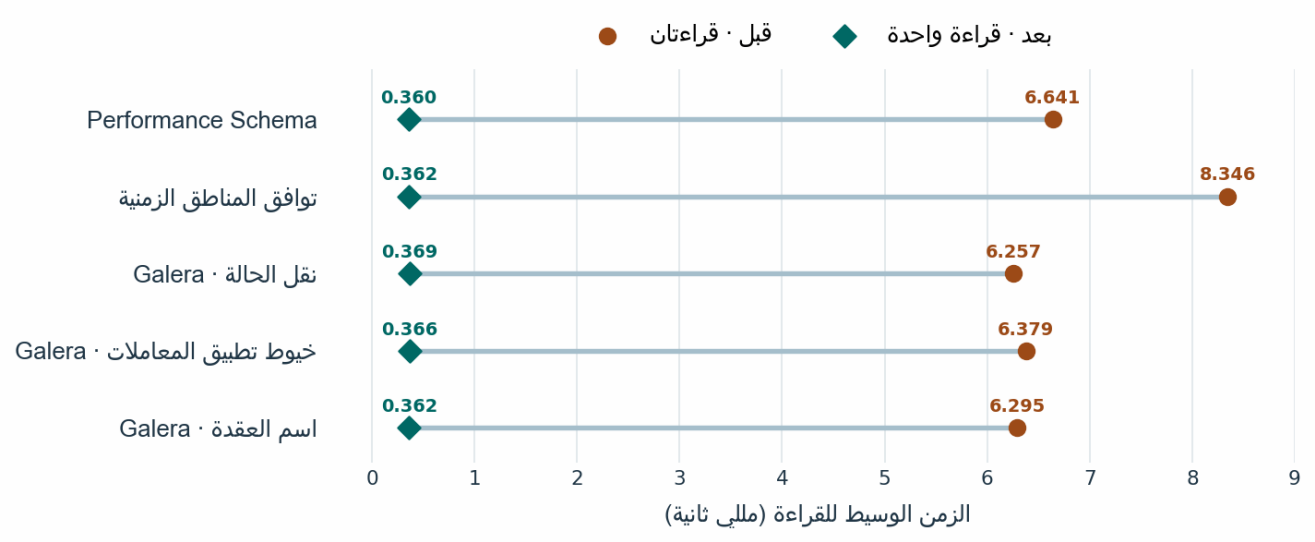
ما الفحوص الخمسة التي اختبرناها؟

عرضت صفحة الملخص خمس عمليات تنفيذ لاستعلام SQL نفسه بقيم مختلفة للبصمة. تحدد كل قيمة حالة مراقَبة. استخرجنا اسم القاعدة من الملاحظة المقابلة. الحروف بين قوسين هي الرموز المستخدمة في ملفات CSV.

الفحص (رمز CSV)	رمز القاعدة	نقاط الإنذار / المجموع
تفعيل Performance Schema (A)	`AdvisorPerformanceSchema`	0 / 5,692

الفحص (رمز CSV)	رمز القاعدة	نقاط الإنذار / المجموع
توافق المناطق الزمنية (B)	`AdvisorTimeZoneAlignment`	5,694 / 5,695
Galera · طريقة نقل الحالة (SST) (C)	`AdvisorWsrepSstMethod`	0 / 5,692
Galera · خيوط تطبيق المعاملات (D)	`AdvisorWsrepSlaveThreads`	0 / 5,692
Galera · اسم العقدة (E)	`AdvisorWsrepNodeName`	0 / 5,692

تعود الملاحظات الخمس المعادة تشغيلها إلى 21 نوفمبر الساعة 22:29:19. عند تلك اللحظة كان فحص المناطق الزمنية وحده في حالة إنذار. يصل كل خط في الرسم بين الوسيط الأصلي ووسيط المسار الذي يبدأ بالبحث عن النقطة المطابقة.



أُجري كل فحص 50 مرة لكل نسخة على الخادم المرصود. يجمع السطر الأخير القياسات الـ250 لكل نسخة. يعرض الجدولان زمن SQL بالميلي ثانية.

إعادة معالجة نقطة معروفة · من قراءتين إلى قراءة واحدة

الفحص	المتوسط قبل	المتوسط بعد	الوسيط قبل	الوسيط بعد
تفعيل Performance Schema	7.171	0.372	6.641	0.360
توافق المناطق الزمنية	8.808	0.400	8.346	0.362
Galera · طريقة نقل الحالة (SST)	7.372	0.575	6.257	0.369

الفحص	المتوسط قبل	المتوسط بعد	الوسيط قبل	الوسيط بعد
Galera · خيوط تطبيق المعاملات	6.970	0.389	6.379	0.366
Galera · اسم العقدة	7.132	0.374	6.295	0.362
الفحوص الخمسة	7.491	0.422	6.645	0.363

إضافة نقطة إلى سجل قائم · من أربع قراءات إلى ثلاث

الفحص	المتوسط قبل	المتوسط بعد	الوسيط قبل	الوسيط بعد
تفعيل Performance Schema	39.832	29.522	29.189	26.465
توافق المناطق الزمنية	41.518	32.461	35.533	29.123
Galera · طريقة نقل الحالة (SST)	39.431	31.138	31.509	28.157
Galera · خيوط تطبيق المعاملات	36.855	31.667	33.790	25.943
Galera · اسم العقدة	39.696	31.043	35.195	28.511
الفحوص الخمسة	39.466	31.166	33.243	27.975

لماذا تكسب إضافة نقطة جديدة وقتاً أقل؟

قد تقع ملاحظة متأخرة بين نقطتين محفوظتين فتغير حدود الحادث. إذا غاب المفتاح المطابق، يبحث التطبيق عن النقطة السابقة واللاحقة بترتيب `(observed_epoch, observed_at)` . يكفي العثور على إحداهما أيضاً لإثبات وجود السجل.

هاتان خطتا البحث عن الجارين لفحص المناطق الزمنية، بمعامل الإضافة المؤرخ في 22 نوفمبر عند 22:29:19. تبقى كلفة هاتين القراءتين بعد حذف البحث عن أحدث تاريخ.

الجار المطلوب	الوصول	الفهرس	الأقسام	الصفوف المقدرة
السابق	`range`	`PRIMARY`	45	8,653

الجار المطلوب	الوصول	الفهرس	الأقسام	الصفوف المقدرة
اللاحق	`range`	`PRIMARY`	45	81

V1 هي النسخة الأصلية. تحذف **V2** البحث عن أحدث تاريخ وتستخدم النقطة المطابقة وجارياها. إذا لم تنجح عمليات البحث الثلاث، تتحقق من وجود حالة حفظها تثبيت قديم. في المجموعة الأولى ارتفع وسيط الاستعادة من 9.665 إلى 12.963 مللي ثانية لأن V2 كررت البحث عن الجيران في سجل تبيّن للتو أنه فارغ.

تعالج **V3**، وهي النسخة المختارة، هذه الحالة مباشرة. إذا كان السجل فارغاً ولم يوجد حادث قديم، تستعيد نقطة الحالة السليمة الوحيدة المحفوظة بتاريخها وقيمة epoch ومحتواها. ثم تتحقق من النقطة الواردة تحت القفل نفسه.

المعاملة الكاملة في المختبر، المتوسط / الوسيط بالمللي ثانية. يشمل توقيت PHP أخذ القفل والقراءات والكتابات و `COMMIT` وتحرير القفل.

العملية	V1 · الأصلية	V2 · حذف القراءة	V3 · الاستعادة المباشرة
إعادة نقطة مسجلة	3.954 / 2.554	1.586 / 0.924	1.518 / 0.904
إضافة إلى سجل موجود	23.907 / 17.297	23.516 / 13.954	22.895 / 14.017
إنشاء أول نقطة	28.373 / 10.477	20.031 / 8.689	22.708 / 8.660
استعادة نقطة قديمة سليمة	23.609 / 10.422	29.137 / 13.423	17.151 / 7.916

يظهر الفرق الأكبر بين V2 وV3 في الاستعادة، أما وسيطا الإضافة فمتقاربان. أعطت النسخ الثلاث القرارات والنقاط والحوادث والفترات والحالات الأخيرة نفسها.

هل تصلح البصمة مفتاحاً أساسياً؟

في 20 نوفمبر أحصينا **543,719 نقطة و9,478 بصمة**. ظهرت بصمة واحدة 83 مرة في ذلك اليوم. تحدد البصمة الفحص، وتحدد الثلاثية `(fingerprint, observed_epoch, observed_at)` قياساً واحداً منه.

قدّر `SHOW INDEX` عدد القيم المختلفة للبادئة الأولى بنحو 3,672,630 وللبادئتين الأطول بنحو 2,448,420. إضافة عمود لا تخفض العدد الفعلي، والانخفاض الظاهر مصدره التقدير. أعطى `PRIMARY KEY(fingerprint)` الخطأ 1503 في النسخة المقسمة يومياً لأن كل مفتاح فريد يجب أن يشمل `observed_at`. وفي النسخة غير المقسمة أعطت النقطة الثانية بالبصمة نفسها خطأ التكرار 1062. MariaDB · SHOW INDEX · MariaDB · Partitioning

طرق أخرى للتحقق من وجود سجل

تشمل الأزمنة الفحوص الخمسة مجتمعة. الخطة المعروضة تخص **توافق المناطق الزمنية**. يشير السهم إلى الرجوع إلى السجل الكامل إذا غابت النتيجة، وتعني الشرطة أن الجدول غير مقسّم. لكل مجموعة قياسها المرجعي الخاص.

المجموعة والاستعلام	الوصول / الفهرس	الأقسام	المتوسط / الوسيط (ms)
I · أحدث تاريخ	`range / PRIMARY`	45	7.577 / 6.660
I · أحدث تاريخ مع `FORCE INDEX (PRIMARY)`	`ref / PRIMARY`	45	4.927 / 4.283
I · `SELECT 1 … LIMIT 1`	`ref / level_neighbour`	45	6.361 / 5.434
I · `SELECT EXISTS(…)`	`ref / level_neighbour`	45	6.870 / 5.625
I · التحقق من الحالة في السجل ثم الرجوع	`const / PRIMARY` + `const / PRIMARY` → `ref / level_neighbour`	— + 45 → 45	4.543 / 3.995
II · أحدث تاريخ	`range / PRIMARY`	45	6.401 / 5.623
II · يوم الملاحظة ثم الرجوع	`ref / level_neighbour_epoch` → `ref / level_neighbour`	1 → 45	3.739 / 3.411
II · قراءة آخر حالة فقط	`const / PRIMARY`	—	0.277 / 0.257

أعادت قراءة آخر حالة وحدها التاريخ نفسه في المقارنات الـ250 كلها. لكن يمكن لهذا الجدول حفظ حالة من إصدار أقدم أو البقاء بعد حذف النقاط بانتهاء مدة الاحتفاظ. لذلك لا يكفي وحده لتحديد ضرورة إعادة بناء السجل.

قراءة استعلامات SQL وخطط جميع البدائل (ملف نصي)

أثر فهرس الوقت والتقسيم

لتقييم البنى نفّذنا الاستعلام الأصلي المرتب على **621,865 نقطة متطابقة** في MariaDB 11.8.10 مع MyRocks. الخطط تخص فحص المناطق الزمنية. الفهرس المقترح هو `when_observed(observed_epoch, observed_at)`.

البنية	الوصول / الفهرس	الأقسام في الخطة	المتوسط / الوسيط (ms)
مفتاح مركّب وأقسام يومية	`range / PRIMARY`	45	612.119 / 549.937
إضافة فهرس الوقت	`range / PRIMARY`	45	637.779 / 525.603
فرض فهرس الوقت	`index / when_observed`	45	64.429 / 48.306
جدول غير مقسّم	`range / PRIMARY`	—	14.052 / 7.676
16 قسماً بحسب البصمة	`ref / PRIMARY`	1	0.786 / 0.410

بعد إضافة الفهرس واصل المحسّن اختيار `PRIMARY` للفحوص الخمسة. عند فرض `when_observed` يمسح الوصول من نوع `index` التواريخ حتى يعثر على البصمة. يحدد التقسيم حسب البصمة قسماً واحداً من 16، لكنه يتطلب تغيير طريقة الحذف اليومي.

خُصص لمختبر معالج CPU واحد وذاكرة كتل MyRocks بسعة 96 ميبيبايت. كان وسيط القياس المرجعي 1.350 مللي ثانية في نسخة أولى من 571,957 نقطة، وبلغ 549.937 مللي ثانية في النسخة الموسعة. سبب الفرق بين المجموعتين لم يُعزل. تصلح هذه القيم للمقارنة داخل المجموعة الواحدة. زمن الإنتاج وكلفة الترحيل يحتاجان إلى قياس مستقل.

طريقة القياس والبيانات المتاحة للتنزيل

ثبّتنا الأمثلة الخمسة التي عرضتها صفحة الملخص معاً. في كل جولة من **50 جولة** خلطنا ترتيب المعاملات والنسخ بذرة ثابتة، فحصلنا على **50 قياساً لكل فحص ونسخة، و250 لكل نسخة**. شملت النتائج المحاولات الأولى وقمم التأخير. استخدمنا على الخادم لقطات للقراءة فقط وعطّلنا ذاكرة نتائج الاستعلامات وأخذنا الأزمنة من `SHOW PROFILES`.

أُجريت معاملات الكتابة على نسخ مستقلة تضم 45 قسماً و28,463 مفتاحاً تاريخياً للفحوص الخمسة ومحتوى تنبيهات اصطناعياً. شملت الاختبارات التأخر وتغير التوقيت الموسمي والتعارض وإعادة المحاولة بعد التراجع. يضم الأرشيف كل قياس وصيغ SQL والخطط الكاملة لكل فحص.

تنزيل القياسات وأسماء الفحوص الخمسة وخطط SQL (ZIP)

ما الذي نختاره لـPmaControl؟

بالعودة إلى فحص المناطق الزمنية، انخفض متوسط زمن القراءات في 50 إعادة **من 8.808 إلى 0.400 مللي ثانية** وانخفض الوسيط **من 8.346 إلى 0.362 مللي ثانية**. يتوقف القرار على الفحص واللحظة والحالة المسجلة. **نعتمد V3 مع المفتاح الأساسي المركّب**. تكفي عملية بحث واحدة للتعرف على الإعادة، ويبقى السجل قادرًا على معالجة النقطة التالية ولو وصلت متأخرة.