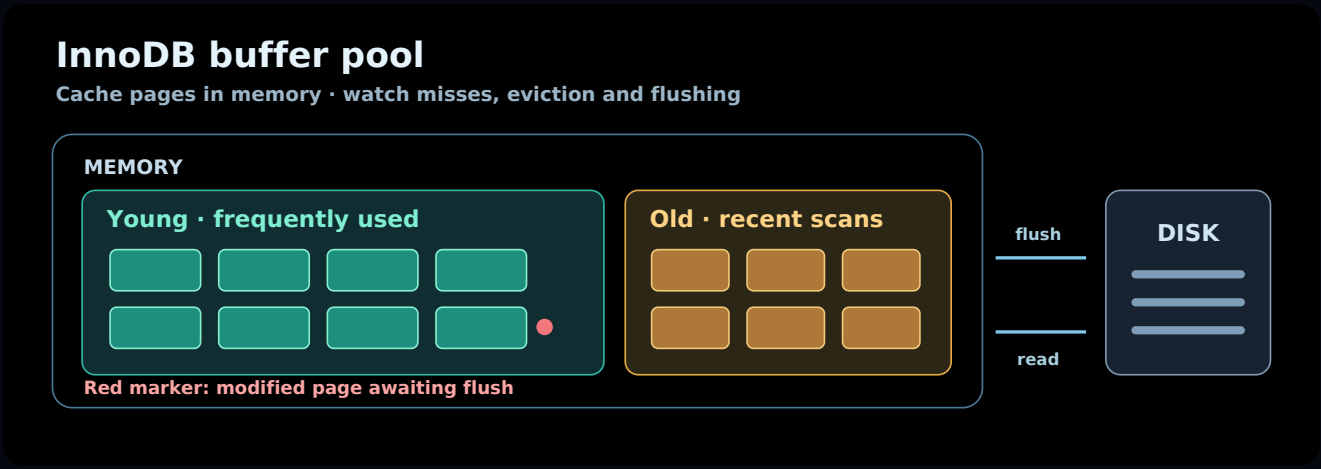
# فهم مخزن InnoDB المؤقت: دليل لتحسين الأداء

**Duthaho** · 28 سبتمبر 2026

MYSQL INNODB BUFFER-POOL PERFORMANCE




يعتمد InnoDB على الذاكرة لتجنب قراءة كل صفحة من القرص في كل مرة. يحتفظ المخزن المؤقت بصفحات البيانات والفهارس المستخدمة حديثًا، وتُعدّل عمليات الكتابة الصفحات في الذاكرة قبل تفريغها إلى القرص. يساعد فهم الصفحات التي تدخل المخزن وتخرج منه، والوقت اللازم لتسخينه، على تفسير الأداء في التشغيل المعتاد وبطء الاستعلامات بعد إعادة التشغيل.

## ما الذي يحتفظ به المخزن المؤقت؟

يحدد `innodb_buffer_pool_size` حجم مساحة الذاكرة المشتركة هذه. يبلغ حجم صفحة InnoDB عادةً 16 كيلوبايت، لكن يمكن ضبط `innodb_page_size` على قيمة مختلفة عند إنشاء المثيل. تتقاسم صفحات البيانات والفهارس المساحة مع البنى الداخلية. إذا كانت الصفحة في الذاكرة فلن يلزم قراءتها من ملف البيانات؛ وإلا فيجب جلبها من التخزين.

قد يفيد مخزن كبير على خادم مخصص لقاعدة البيانات، لكن لا توجد نسبة ثابتة من الذاكرة تناسب كل الأنظمة. اترك مساحة لنظام التشغيل والاتصالات والعمليات المؤقتة ومحركات التخزين الأخرى والخدمات الموجودة على الخادم نفسه. مخزن بحجم 48 جيبيبايت على خادم ذاكرته 64 جيبيبايت هو **مثال ينبغي اختباره**، وليس توصية عامة.

قِس القراءات خلال الفترة الزمنية نفسها بدل الاعتماد على نسبة تراكمية منذ تشغيل الخادم:

```
SHOW GLOBAL STATUS WHERE Variable_name IN (
  'Innodb_buffer_pool_read_requests',
  'Innodb_buffer_pool_reads',
  'Innodb_buffer_pool_pages_dirty',
  'Innodb_buffer_pool_wait_free'
);
```

نسبة عدم العثور على الصفحة خلال فترة محددة تساوي `ΔInnodb_buffer_pool_reads / ΔInnodb_buffer_pool_read_requests`. إذا ارتفعت هذه النسبة مع القراءات الفعلية وزمن الاستعلامات، فقد لا تعود مجموعة البيانات النشطة مخدومة جيداً من الذاكرة. لا تثبت نسبة نجاح مرتفعة وحدها أن الاستعلام سريع. يساعد دد الصفحات المعدلة و `Innodb_buffer_pool_wait_free` على فهم ضغط التفريغ إلى القرص.

## لماذا لا يطرد المسح الكبير كل الصفحات كثيرة الاستخدام بالضرورة؟

يستخدم InnoDB قائمة LRU معدلة. تدخل الصفحات الجديدة إلى القسم **old**، ويمكن للصفحات كثيرة الاستخدام الانتقال إلى القسم **young**. يخفف الإدراج في منتصف القائمة من إزاحة مجموعة العمل النشطة عند إجراء مسح للجدول لمرة واحدة.

يضبط `innodb_old_blocks_pct` حصة قسم old من القائمة (القيمة الافتراضية 37 في MySQL 8.4). ويحدد `innodb_old_blocks_time` مهلة ترقية الصفحة الجديدة (القيمة الافتراضية 1000 مللي ثانية في MySQL 8.4). قد تحمي مهلة أطول صفحات OLTP أثناء التقارير الكبيرة، لكنها قد تؤخر ترقية صفحات مفيدة بالفعل. وضع القيمة صفراً يلغي المهلة، وليس علاجاً عاماً لأحمال القراءة. قِس الحمل الفعلي قبل تغيير أي من الإعدادين.

## تسخين المخزن بعد إعادة التشغيل أو التحويل إلى خادم آخر

بعد إعادة التشغيل يبدأ المخزن بارداً. قد تقرأ الطلبات الأولى ملفات البيانات بكثرة حتى ولو لم تتغير الاستعلامات أو الفهارس. يستطيع InnoDB حفظ **معرّفات** الصفحات عند الإيقاف ثم إعادة تحميلها عند البدء باستخدام `innodb_buffer_pool_dump_at_shutdown` و `innodb_buffer_pool_load_at_startup`. يحدد `innodb_buffer_pool_dump_pct` نسبة الصفحات الأكثر استخداماً التي تُسجّل. لا يحتوي ملف التفريغ على الصفحات نفسها؛ ما زالت إعادة التحميل تحتاج إلى قراءات من القرص.

يمكن أيضاً ترك حركة المرور العادية لتسخين المخزن، أو تنفيذ استعلامات ممثلة ومحدودة النطاق، أو تسخين نسخة متماثلة قبل نقل الحركة إليها. قد تتنافس عملية التحميل المسبق مع الإنتاج على عمليات الإدخال والإخراج، لذا اضبط سرعتها وراقب زمن الاستجابة. امتلاء المخزن بنسبة 80% لا يعني بالضرورة أن صفحات الاستعلامات المهمة أصبحت جاهزة. قيّم الجاهزية من زمن الاستجابة والقراءات الفعلية لطلبات ممثلة، ومن حالة تحميل المخزن عند استخدام هذه الآلية.

## علاقة الكتابة وredo بالبنى الأخرى

عندما تعدل معاملة صفحةً، تصبح نسختها في الذاكرة **صفحة متسخة**. يسجل InnoDB بيانات redo ليتمكن من إعادة تطبيق التغييرات المثبتة بعد العطل؛ وتكتب عمليات الخلفية لاحقاً الصفحات المعدلة إلى ملفات البيانات. يؤدي ذلك إلى تقدم نقطة التحقق. لكل من سجل redo والمخزن المؤقت وظيفة مختلفة: الهدف الشديد لتقليل الصفحات المتسخة قد يزيد عمليات الإدخال والإخراج، بينما قد يسبب التفريغ البطيء ضغطاً لاحقاً. اختر مستوى المتانة المناسب؛ عادةً يُستخدم `innodb_flush_log_at_trx_commit=1` عندما يجب أن تنجو كل معاملة مثبتة من تعطل الخادم.

يمكن لـ **change buffer** تأجيل تعديلات صفحات الفهارس الثانوية غير الفريدة غير الموجودة حالياً في الذاكرة. قد يقلل ذلك عمليات الإدخال والإخراج العشوائية لبعض أحمال الكتابة في MySQL، مقابل عمليات دمج لاحقة واستهلاك للذاكرة. ويمكن لـ **adaptive hash index** تسريع عمليات البحث المتكررة، لكنه يستهلك الذاكرة أيضاً وقد يسبب تزاحماً بين الخيوط. الـ تعني إتاحة أي منهما أنه مفيد تلقائياً: تحقق من دعمه في إصدارك، وراقب العدادات، واختبره بحركة ممثلة قبل تغيير الإعدادات. وعلى وجه الخصوص، تستخدم MariaDB منذ الإصدار 10.5 مثيلاً واحداً للمخزن، ولذلك لا تنطبق عليها نصيحة ضبط `innodb_buffer_pool_instances=8`.

## خطوات عملية للضبط

في نظام طلبات مزدحم، ابدأ بمجموعة العمل الفعلية وميزانية الذاكرة. سجّل زمن الاستعلامات والقراءات الفعلية والصفحات المتسخة والإدخال والإخراج في الحمل العادي وحمل الذروة. إذا زادت قراءات القرص عند امتلاء المخزن، حسّن الفهارس وطريقة وصول SQL إلى البيانات أولاً، ثم اختبر مخزناً أكبر مع مراقبة إجمالي ذاكرة الخادم. إذا أزاح مسح تحليلي الصفحات النشطة، اختبر إعدادات قسم old. وإذا ارتفع زمن الاستجابة فقط بعد إعادة التشغيل أو ترقية نسخة متماثلة، فاختبر dump/load وسخّن الخادم الأساسي المقبل قبل توجيه حركة الإنتاج إليه.

في MySQL 8.4 قد تقلل مثيلات المخزن المتعددة التزاحم في المخازن التي يبلغ حجمها عدة جيبيبايت، لكن ينبغي تحديد عددها استنادًا إلى القياس وترك نحو 1 جيبيبايت أو أكثر لكل مثيل. تعد نسبة نجاح عامة مثل 99% مؤشرًا تشخيصيًا، وليست هدفًا للخدمة بمفردها. قارن التغييرات بالحمل نفسه، مع عرض مكاسب السرعة وأي تراجع في استهلاك الذاكرة أو الإدخال والإخراج.

## الخلاصة

المخزن المؤقت ليس مجرد مساحة تخزين مؤقت كبيرة. تحمي سياسة إدخال الصفحات البيانات كثيرة الاستخدام، ويحدد التسخين زمن الاستجابة بعد إعادة التشغيل، ويربط التفريغ عمليات الكتابة في الذاكرة بالتخزين الدائم. اضبطه وفق قياسات حملك الفعلي، مع هامش كافٍ من الذاكرة، بدل اتباع وصفة ثابتة لكل البيئات.

تمت تهيئة هذا النص وترجمته بإذن المؤلف اعتمادًا على مقال Duthaho الأصلي في Medium، المنشور أول مرة في 22 أغسطس 2025. تمت مراجعة التفاصيل المرتبطة بالإصدارات في توثيق MySQL 8.4 للمخزن المؤقت وتوثيق MariaDB.