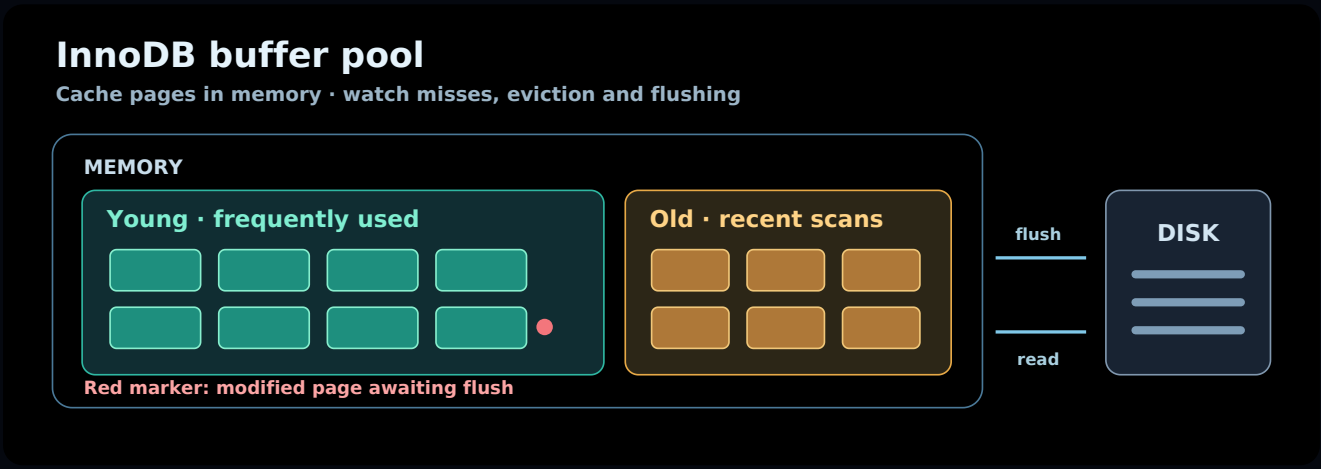
Буферный пул InnoDB: устройство и настройка производительности

Duthaho · September 28, 2026

MYSQL INNODB BUFFER-POOL PERFORMANCE



InnoDB использует оперативную память, чтобы не читать каждую страницу с диска. Буферный пул хранит недавно использованные страницы данных и индексов, а операции записи изменяют их в памяти до сброса на диск. Понимание того, какие страницы попадают в пул и покидают его, а также сколько времени занимает прогрев, помогает объяснить производительность в обычном режиме и задержки после перезапуска.

Что хранится в буферном пуле

`innodb_buffer_pool_size` задаёт размер этой общей области памяти. Обычно страницы InnoDB имеют размер 16 КиБ, но `innodb_page_size` можно задать иначе при создании экземпляра. Страницы данных и индексов делят место с внутренними структурами. Если страница уже в памяти, читать файл данных не нужно; при её отсутствии потребуется чтение с накопителя.

На выделенном сервере базы данных большой пул может быть полезен, однако единого процента оперативной памяти для всех случаев нет. Оставьте память операционной системе, соединениям, временным операциям, другим движкам и соседним службам. Пул размером 48 ГиБ на сервере с 64 ГиБ памяти — **пример для проверки**, а не универсальная рекомендация.

Измеряйте чтения за один и тот же интервал, а не полагайтесь на накопленный с запуска процент:

```
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'
);
```

Доля промахов за интервал равна `ΔInnodb_buffer_pool_reads / ΔInnodb_buffer_pool_read_requests`. Если она растёт вместе с физическими чтениями и задержкой запросов, рабочий набор данных, возможно, уже плохо помещается в памяти. Высокая доля попаданий сама по себе не доказывает, что запрос выполняется быстро. Число грязных страниц и `Innodb_buffer_pool_wait_free` помогают оценить нагрузку на сброс страниц.

Почему большой скан не обязательно вытеснит все горячие страницы

InnoDB применяет модифицированный список LRU. Новые страницы попадают в часть **old**, а часто используемые могут перейти в часть **young**. Вставка в середине списка защищает часть активного рабочего набора от однократного сканирования таблицы.

`innodb_old_blocks_pct` управляет долей части old (по умолчанию 37 в MySQL 8.4). `innodb_old_blocks_time` задаёт задержку перед повышением новой страницы (по умолчанию 1000 мс в MySQL 8.4). Более долгий период может защитить страницы OLTP во время большого отчёта, но также задержать продвижение действительно полезных страниц. Нулевое значение убирает задержку; это не универсальное решение для нагрузки на чтение. Перед изменением параметров измерьте реальную нагрузку.

Прогрев после перезапуска или переключения

После перезапуска кеш холодный. Первые запросы могут интенсивно читать файлы данных, хотя SQL и индексы не изменились. InnoDB умеет сохранять **идентификаторы** страниц при остановке и загружать их при запуске с помощью

`innodb_buffer_pool_dump_at_shutdown` и `innodb_buffer_pool_load_at_startup`. `innodb_buffer_pool_dump_pct` определяет долю наиболее востребованных страниц в списке. Самих страниц в дампе нет: при запуске данные всё равно читаются с диска.

Другие варианты — позволить обычному трафику прогреть кеш, выполнить ограниченный набор репрезентативных запросов или подготовить реплику до перевода на неё трафика. Предварительная загрузка может конкурировать с рабочей нагрузкой за операции ввода-вывода, поэтому ограничивайте её скорость и наблюдайте за задержками. Заполнение пула на 80% ещё не означает готовность важных запросов. Оценивайте готовность по задержкам и физическим чтениям репрезентативных запросов и, если применимо, по состоянию загрузки пула.

Записи, redo и другие структуры

Когда транзакция меняет страницу, её копия в кеше становится **грязной**. InnoDB записывает данные в журнал redo, чтобы после сбоя воспроизвести подтверждённые изменения; фоновые процессы позже сбрасывают изменённые страницы в файлы данных. Это продвигает контрольную точку. Журнал redo и буферный пул решают разные задачи: слишком агрессивная цель по грязным страницам увеличивает ввод-вывод, а слишком медленный сброс может позднее создать перегрузку. Выбирайте надёжность согласно требованиям: `innodb_flush_log_at_trx_commit=1` обычно используют, если каждая подтверждённая транзакция должна пережить сбой сервера.

Change buffer может откладывать изменения страниц неуникальных вторичных индексов, которых сейчас нет в памяти. На некоторых нагрузках записи MySQL это сокращает случайные операции ввода-вывода ценой последующего слияния и расхода памяти. **Adaptive hash index** может ускорять повторяющиеся поиски, но также расходует память и иногда вызывает конкуренцию потоков. Ни одна из этих функций не полезна автоматически: проверьте поддержку в своей версии, изучите счётчики и испытайте изменения на репрезентативной нагрузке. В частности, начиная с MariaDB 10.5 используется один экземпляр буферного пула, поэтому совет установить `innodb_buffer_pool_instances=8` к ней неприменим.

Практический порядок настройки

Для загруженной системы заказов начните с реального рабочего набора и бюджета памяти. Зафиксируйте задержки запросов, физические чтения, число грязных страниц и ввод-вывод при обычной и пиковой нагрузке. Если при заполненном пуле увеличивается число чтений с диска, сначала улучшите индексы и способы доступа к данным, затем проверьте больший пул, наблюдая за общей памятью сервера. Если аналитический скан вытесняет горячие страницы, испытайте параметры части old. Если задержки появляются лишь после перезапуска или повышения реплики, проверьте dump/load и прогрейте будущий основной узел до подачи на него рабочего трафика.

В MySQL 8.4 несколько экземпляров пула могут снизить конкуренцию в пулах объёмом несколько гигабайт, но их число должно опираться на измерения и оставлять примерно 1 ГиБ или больше на экземпляр. Общая цель попаданий вроде 99% — диагностическая подсказка, а не самостоятельный показатель качества сервиса. Сравнивайте изменения на одинаковой нагрузке, учитывая ускорение и возможный рост потребления памяти или ввода-вывода.

Вывод

Буферный пул — больше чем большой кеш. Правила размещения защищают востребованные страницы, прогрев влияет на задержки после перезапуска, а сброс связывает записи в памяти с надёжным хранением. Настраивайте пул по результатам измерений своей нагрузки, оставляя запас памяти вместо следования универсальному рецепту.

Материал адаптирован и переведён с разрешения автора по статье Duthaho на Medium, впервые опубликованной 22 августа 2025 года. Сведения, зависящие от версии, сверены с документацией MySQL 8.4 и документацией MariaDB.