

Метрики Tarantool в VK WorkSpace

Инструкция для администраторов

Оглавление

Общие метрики	3
Память (общая)	3
Выделение памяти (Slab allocator)	4
Спейсы (Spaces)	5
Сеть (Network)	5
Файберы (Fibers)	6
Операции (Operations)	7
Репликация (Replication)	8
Среда выполнения (Runtime)	8
Метрики LuaJIT	9
Метрики CPU	10
Vinyl	11
Метрики цикла событий (Event loop)	14
Синхро (Synchro)	14
Выборы лидера (Election)	15
Memtx (MVCC)	15
Кортежи (Tuples)	16
Статистика Read View	17
Конфигурация Tarantool	18

Общие метрики

Общая информация об инстансе:

Метрика	Описание
<code>tnt_cfg_current_time</code>	Системное время инстанса в формате Unix timestamp.
<code>tnt_info_uptime</code>	Время в секундах с момента запуска инстанса.
<code>tnt_read_only</code>	Указывает, находится ли инстанс в режиме "только для чтения" (1 если да, 0 если нет).

Память (общая)

Следующие метрики дают картину использования памяти процессом Tarantool.

Метрика	Описание
<code>tnt_info_memory_cache</code>	Количество байт в кэше, используемом для хранения кортежей движка vinyl.
<code>tnt_info_memory_data</code>	Количество байт, используемых для хранения пользовательских данных (кортежей) в движке memtx и на уровне 0 движка vinyl, без учета фрагментации памяти.
<code>tnt_info_memory_index</code>	Количество байт, используемых для индексации пользовательских данных. Включает деревья памяти memtx и vinyl, страничный индекс vinyl и фильтры Блума vinyl.
<code>tnt_info_memory_lua</code>	Количество байт, используемых средой выполнения Lua. Мониторинг этой метрики может предотвратить переполнение памяти.
<code>tnt_info_memory_net</code>	Количество байт, используемых для буферов сетевого ввода/вывода.
<code>tnt_info_memory_tx</code>	Количество байт, используемых активными транзакциями. Для движка vinyl это общий размер всех выделенных объектов (<code>struct</code>

Метрика	Описание
	<code>txv</code> , <code>struct vy_tx</code> , <code>struct vy_read_interval</code>) и закрепленных за ними кортежей.

Выделение памяти (Slab allocator)

Отчет об использовании памяти слэб-аллокатором. Слэб-аллокатор является основным для хранения кортежей. **Доступная память, байты:**

Метрика	Описание
<code>tnt_slab_quota_size</code>	Объем памяти, доступный для хранения кортежей и индексов. Равен параметру <code>memtx_memory</code> .
<code>tnt_slab_arena_size</code>	Общий объем памяти, доступный для хранения как кортежей, так и индексов. Включает выделенные, но свободные в данный момент слэбы.
<code>tnt_slab_items_size</code>	Общий объем памяти, доступный для хранения только кортежей (не индексов). Включает выделенные, но свободные в данный момент слэбы.

Использование памяти, байты:

Метрика	Описание
<code>tnt_slab_quota_used</code>	Объем памяти, уже зарезервированный слэб-аллокатором.
<code>tnt_slab_arena_used</code>	Эффективно используемая память для хранения как кортежей, так и индексов. Без учета выделенных, но свободных слэбов.
<code>tnt_slab_items_used</code>	Эффективно используемая память для хранения только кортежей (не индексов). Без учета выделенных, но свободных слэбов.

Использование памяти, %:

Метрика	Описание
<code>tnt_slab_quota_used_ratio</code>	$tnt_slab_quota_used / tnt_slab_quota_size$

Метрика	Описание
<code>tnt_slab_arena_used_ratio</code>	<code>tnt_slab_arena_used / tnt_slab_arena_size</code>
<code>tnt_slab_items_used_ratio</code>	<code>tnt_slab_items_used / tnt_slab_items_size</code>

Спейсы (Spaces)

Следующие метрики предоставляют информацию о каждом отдельном спейсе в инстансе Tarantool.

Метрика	Описание
<code>tnt_space_len</code>	Количество записей в спейсе. Всегда имеет метки: <code>{name="test", engine="memtx"}</code> , где <code>name</code> — имя спейса, <code>engine</code> — его движок.
<code>tnt_space_bsize</code>	Общее количество байт во всех кортежах спейса. Всегда имеет метки: <code>{name="test", engine="memtx"}</code> .
<code>tnt_space_index_bsize</code>	Общее количество байт, занимаемых индексом. Всегда имеет метки: <code>{name="test", index_name="pk"}</code> , где <code>name</code> — имя спейса, <code>index_name</code> — имя индекса.
<code>tnt_space_total_bsize</code>	Общий размер кортежей и всех индексов в спейсе. Всегда имеет метки: <code>{name="test", engine="memtx"}</code> .
<code>tnt_vinyl_tuples</code>	Общее количество кортежей для vinyl. Всегда имеет метки: <code>{name="test", engine="vinyl"}</code> . По умолчанию эта метрика отключена и может быть включена только глобальной переменной: <code>rawset(_G, 'include_vinyl_count', true)</code> .

Сеть (Network)

Статистика сетевой активности. Метрики для мониторинга сетевой нагрузки, пиков использования и потерь трафика.

Метрика	Описание
<code>tnt_net_sent_total</code>	Байт, отправленных из инстанса по сети с момента его запуска.

Метрика	Описание
<code>tnt_net_received_total</code>	Байт, полученных инстансом с момента запуска.
<code>tnt_net_connections_total</code>	Количество входящих сетевых соединений с момента запуска инстанса.
<code>tnt_net_connections_current</code>	Количество активных сетевых соединений.
<code>tnt_net_requests_total</code>	Количество сетевых запросов, обработанных инстансом с момента запуска.
<code>tnt_net_requests_current</code>	Количество ожидающих обработки сетевых запросов.
<code>tnt_net_requests_in_progress_total</code>	Общее количество запросов, обработанных потоком <code>tx</code> (транзакций).
<code>tnt_net_requests_in_progress_current</code>	Количество запросов, обрабатываемых в данный момент в потоке <code>tx</code> .
<code>tnt_net_requests_in_stream_total</code>	Общее количество запросов, помещенных в очереди потоков (стримов) за все время.
<code>tnt_net_requests_in_stream_current</code>	Количество запросов, ожидающих в очередях потоков в данный момент.

Начиная с Tarantool 2.10, каждая сетевая метрика имеет метку `thread`, показывающую статистику по потокам.

Файберы (Fibers)

Статистика по файберам. Если ваше приложение создает много файберов, вы можете использовать эти метрики для мониторинга их количества и использования памяти.

Метрика	Описание
<code>tnt_fiber_amount</code>	Количество файберов.
<code>tnt_fiber_csw</code>	Общее количество переключений контекста файберов.

Метрика	Описание
<code>tnt_fiber_memalloc</code>	Объем памяти, зарезервированной для фиберов.
<code>tnt_fiber_memused</code>	Объем памяти, используемой фиберами.

Операции (Operations)

Вы можете собирать данные об `iproto`-запросах, обработанных инстансом, и агрегировать их по типу запроса. Это поможет узнать, какие операции ваши клиенты выполняют чаще всего.

Метрика	Описание
<code>tnt_stats_op_total</code>	Общее количество вызовов с момента запуска сервера.

Для различия между типами запросов эта метрика имеет метку `operation`. Например: `{operation="select"}`.

Возможные типы операций (`operation`):

Метка	Описание
<code>auth</code>	Запросы аутентификации.
<code>call</code>	Запросы на выполнение хранимых процедур.
<code>delete</code>	Запросы на удаление.
<code>error</code>	Запросы, завершившиеся ошибкой.
<code>eval</code>	Вызовы для выполнения кода Lua.
<code>execute</code>	Запросы на выполнение SQL.
<code>insert</code>	Запросы на вставку.
<code>prepare</code>	Запросы на подготовку SQL.
<code>replace</code>	Запросы на замену.

Метка	Описание
<code>select</code>	Запросы на выборку.
<code>update</code>	Запросы на обновление.
<code>upsert</code>	Запросы на вставку или обновление.

Репликация (Replication)

Предоставляет текущий статус репликации.

Метрика	Описание
<code>tnt_info_lsn</code>	LSN (Log Sequence Number) инстанса.
<code>tnt_info_vclock</code>	Номер LSN в векторных часах (vclock). Всегда имеет метку <code>{id="id"}</code> , где <code>id</code> — номер инстанса в наборе реплик.
<code>tnt_replication_lsn</code>	LSN инстанса Tarantool. Всегда имеет метки <code>{id="id", type="type"}</code> , где <code>id</code> — номер инстанса, <code>type</code> — <code>master</code> или <code>replica</code> .
<code>tnt_replication_lag</code>	Значение лага репликации в секундах. Всегда имеет метки <code>{id="id", stream="stream"}</code> , где <code>id</code> — номер инстанса, <code>stream</code> — <code>downstream</code> или <code>upstream</code> .
<code>tnt_replication_status</code>	Эта метрика равна 1, когда статус репликации — «follow», и 0 в противном случае. Всегда имеет метки <code>{id="id", stream="stream"}</code> .

Среда выполнения (Runtime)

Метрика	Описание
<code>tnt_runtime_lua</code>	Размер сборщика мусора Lua в байтах.
<code>tnt_runtime_used</code>	Количество байт, используемых средой выполнения Lua.

Метрика	Описание
<code>tnt_runtime_tuple</code>	Количество байт, используемых кортежами (кроме кортежей, принадлежащих memtx и vinyl).

Метрики LuaJIT

Метрики LuaJIT дают представление о работе сборщика мусора Lua. Доступны в Tarantool 2.6 и новее.

Общие JIT-метрики:

Метрика	Описание
<code>lj_jit_snap_restore_total</code>	Общее количество восстановлений снимков.
<code>lj_jit_trace_num</code>	Количество JIT-трейсов.
<code>lj_jit_trace_abort_total</code>	Общее количество прерванных трейсов.
<code>lj_jit_mcode_size</code>	Общий размер выделенных областей машинного кода.

Строки JIT:

Метрика	Описание
<code>lj_strhash_hit_total</code>	Количество интернированных строк.
<code>lj_strhash_miss_total</code>	Общее количество выделений строк.

Шаги GC:

Метрика	Описание
<code>lj_gc_steps_atomic_total</code>	Количество инкрементальных шагов GC (состояние atomic).
<code>lj_gc_steps_sweepstring_total</code>	Количество инкрементальных шагов GC (состояние sweepstring).

Метрика	Описание
<code>lj_gc_steps_finalize_total</code>	Количество инкрементальных шагов GC (состояние finalize).
<code>lj_gc_steps_sweep_total</code>	Количество инкрементальных шагов GC (состояние sweep).
<code>lj_gc_steps_propagate_total</code>	Количество инкрементальных шагов GC (состояние propagate).
<code>lj_gc_steps_pause_total</code>	Количество инкрементальных шагов GC (состояние pause).

Выделения (Allocations):

Метрика	Описание
<code>lj_gc_strnum</code>	Количество выделенных строковых объектов.
<code>lj_gc_tabnum</code>	Количество выделенных табличных объектов.
<code>lj_gc_cdatanum</code>	Количество выделенных cdata-объектов.
<code>lj_gc_udatanum</code>	Количество выделенных udata-объектов.
<code>lj_gc_freed_total</code>	Общий объем освобожденной памяти.
<code>lj_gc_memory</code>	Текущая выделенная память Lua.
<code>lj_gc_allocated_total</code>	Общий объем выделенной памяти.

Метрики CPU

Следующие метрики предоставляют статистику использования CPU. Доступны только на Linux.

Метрика	Описание
<code>tnt_cpu_number</code>	Общее количество процессоров, сконфигурированных операционной системой.

Метрика	Описание
<code>tnt_cpu_time</code>	Процессорное время хоста.
<code>tnt_cpu_thread</code>	Процессорное время потока Tarantool. Всегда имеет метки: <code>{kind="user", thread_name="tarantool", thread_pid="pid", file_name="init.lua"}</code> , где <code>kind</code> может быть <code>user</code> или <code>system</code> , <code>thread_name</code> — <code>tarantool</code> , <code>wal</code> , <code>iproto</code> или <code>coio</code> , <code>file_name</code> — имя входного файла.

Существуют также две кроссплатформенные метрики, которые можно получить с помощью вызова `getrusage()`:

Метрика	Описание
<code>tnt_cpu_user_time</code>	Процессорное время пользователя Tarantool.
<code>tnt_cpu_system_time</code>	Системное процессорное время Tarantool.

Vinyl

Метрики Vinyl предоставляют статистику движка vinyl.

Диск (Disk)

Метрики диска используются для мониторинга общего объема данных на диске.

Метрика	Описание
<code>tnt_vinyl_disk_data_size</code>	Объем данных в байтах, хранящихся в файлах <code>.run</code> , расположенных в <code>vinyl_dir</code> .
<code>tnt_vinyl_disk_index_size</code>	Объем данных в байтах, хранящихся в файлах <code>.index</code> , расположенных в <code>vinyl_dir</code> .

Регулятор (Regulator)

Регулятор vinyl решает, когда начинать действия ввода-вывода на диск. Он группирует действия в пакеты для повышения эффективности.

Метрика	Описание
<code>tnt_vinyl_regulator_dump_bandwidth</code>	Расчетная средняя скорость дампа, байт в секунду. Начальное значение — 10485760 (10 МБ/с). Пересчитывается в зависимости от фактической скорости. Для оценки используются только значимые дампы размером более 1 МБ.
<code>tnt_vinyl_regulator_write_rate</code>	Фактическая средняя скорость выполнения операций записи, байт в секунду. Рассчитывается как скользящее среднее за 5 секунд. Если значение метрики постепенно снижается, это может указывать на проблемы с диском.
<code>tnt_vinyl_regulator_rate_limit</code>	Ограничение скорости записи, байт в секунду. Регулятор накладывает ограничение на транзакции на основе наблюдаемой производительности дампа/компрессии. Если значение метрики падает примерно до 10^5 , это указывает на проблемы с диском или планировщиком.
<code>tnt_vinyl_regulator_dump_watermark</code>	Максимальный объем памяти в байтах, используемый для хранения LSM-дерева <code>vinyl</code> в памяти. При достижении этого максимума должен произойти дамп. Значение немного меньше объема памяти, выделенного для деревьев <code>vinyl</code> , который отражается в параметре <code>vinyl_memory</code> .
<code>tnt_vinyl_regulator_blocked_writers</code>	Количество файберов, заблокированных в ожидании квоты памяти уровня 0 <code>vinyl</code> .

Транзакционная активность (Transactional activity)

Метрика	Описание
<code>tnt_vinyl_tx_commit</code>	Счетчик коммитов (успешных завершений транзакций). Включает неявные коммиты (например, любая операция вставки вызывает коммит, если она не внутри блока <code>box.begin()</code> – <code>box.commit()</code>).
<code>tnt_vinyl_tx_rollback</code>	Счетчик откатов (неуспешных завершений транзакций). Это не просто количество явных <code>box.rollback()</code> , но и запросы, завершившиеся ошибками.
<code>tnt_vinyl_tx_conflict</code>	

Метрика	Описание
	Счетчик конфликтов, вызвавших откат транзакций. Если отношение <code>tnt_vinyl_tx_conflict / tnt_vinyl_tx_commit</code> выше 5%, это указывает на нездоровое состояние <code>vinyl</code> .
<code>tnt_vinyl_tx_read_views</code>	Текущее количество <code>read view</code> (представлений чтения) — транзакций, которые временно перешли в состояние "только чтение" для избежания конфликта. Обычно значение равно 0. Если оно долго остается ненулевым, это указывает на утечку памяти.

Память (Memory)

Следующие метрики показывают области памяти, используемые `vinyl` для кэшей и буферов записи.

Метрика	Описание
<code>tnt_vinyl_memory_tuple_cache</code>	Объем памяти в байтах, используемый для хранения кортежей (данных) в кэше.
<code>tnt_vinyl_memory_level0</code>	Объем памяти уровня 0 (L0) в байтах. L0 — область, которую <code>vinyl</code> может использовать для хранения LSM-дерева в памяти. Наблюдая за этой метрикой, можно увидеть, когда L0 приближается к своему максимуму (<code>tnt_vinyl_regulator_dump_watermark</code>), после чего произойдет дамп. Можно ожидать, что сразу после завершения дампа <code>L0 = 0</code> .
<code>tnt_vinyl_memory_page_index</code>	Объем памяти в байтах, используемый для хранения индексов. Если значение метрики близко к <code>vinyl_memory</code> , это указывает на неверный выбор <code>vinyl_page_size</code> .
<code>tnt_vinyl_memory_bloom_filter</code>	Объем памяти в байтах, используемый фильтрами Блума.
<code>tnt_vinyl_memory_tuple</code>	Общий размер памяти в байтах, занимаемый кортежами <code>vinyl</code> . Включает кэшированные кортежи и кортежи, закрепленные миром Lua.

Планировщик (Scheduler)

Планировщик `vinyl` вызывает регулятор и обновляет связанные переменные. Это происходит раз в секунду.

Метрика	Описание
<code>tnt_vinyl_scheduler_tasks</code>	Количество задач дампа/компрессии планировщика. Всегда имеет метку <code>{status = <status_value>}</code> , где <code>status_value</code> может быть <code>inprogress</code> (выполняется), <code>completed</code> (успешно завершена) или <code>failed</code> (прервана из-за ошибок).
<code>tnt_vinyl_scheduler_dump_time</code>	Общее время в секундах, затраченное всеми рабочими потоками на выполнение дампов.
<code>tnt_vinyl_scheduler_dump_total</code>	Счетчик завершенных дампов.

Метрики цикла событий (Event loop)

Информация о потоке цикла событий `tx`:

Метрика	Описание
<code>tnt_ev_loop_time</code>	Время цикла событий (мс).
<code>tnt_ev_loop_prolog_time</code>	Время пролога цикла событий (мс).
<code>tnt_ev_loop_epilog_time</code>	Время эпилога цикла событий (мс).

Синхро (Synchro)

Показывает текущее состояние синхронной репликации.

Метрика	Описание
<code>tnt_synchro_queue_owner</code>	ID инстанса текущего мастера синхронной репликации.
<code>tnt_synchro_queue_term</code>	Текущий терм очереди.
<code>tnt_synchro_queue_len</code>	Сколько транзакций в данный момент ожидают подтверждений.
<code>tnt_synchro_queue_busy</code>	

Метрика	Описание
	Обрабатывает ли очередь какие-либо системные записи (CONFIRM/ROLLBACK/PROMOTE/DEMOTE).

Выборы лидера (Election)

Показывает текущее состояние узла набора реплик в отношении выборов лидера.

Метрика	Описание
<code>tnt_election_state</code>	Состояние (режим) узла в процессе выборов. Когда выборы включены, узел доступен для записи только в состоянии лидера. Возможные значения: <ul style="list-style-type: none"> 0 (follower): все не-лидеры называются подписчиками. 1 (candidate): узлы, начинающие новый раунд выборов, называются кандидатами. 2 (leader): узел, собравший кворум голосов, становится лидером.
<code>tnt_election_vote</code>	ID узла, за который голосует текущий узел. Если значение равно 0, это означает, что узел еще не голосовал в текущем терме.
<code>tnt_election_leader</code>	ID узла-лидера в текущем терме. Если значение равно 0, это означает, что узел не знает, какой узел является лидером в текущем терме.
<code>tnt_election_term</code>	Текущий терм выборов.
<code>tnt_election_leader_idle</code>	Время в секундах с момента последнего взаимодействия с известным лидером.

Memtx (MVCC)

Статистика памяти для многоверсионного управления конкурентным доступом (MVCC) в memtx. Менеджер транзакций состоит из двух частей: сами транзакции (секция TXN) и MVCC.

TXN

Метрика	Описание
<code>tnt_memtx_txn_statements</code>	Операторы (statements) транзакции. Всегда имеет метку <code>{kind="..."}</code> с возможными значениями: <code>total</code> : количество байт, выделенных под операторы всех текущих транзакций. <code>average</code> : среднее количество байт, используемое транзакциями для операторов (<code>txn.statements.total bytes</code> / количество открытых транзакций). <code>max</code> : максимальное количество байт, используемое одной из текущих транзакций для операторов.
<code>tnt_memtx_txn_user</code>	Память, выделенная через функцию <code>box_txn_alloc()</code> для текущей транзакции. Всегда имеет метку <code>{kind="..."}</code> с возможными значениями <code>total</code> , <code>average</code> , <code>max</code> (аналогично выше).
<code>tnt_memtx_txn_system</code>	Внутренние механизмы: логи, точки сохранения. Всегда имеет метку <code>{kind="..."}</code> с возможными значениями <code>total</code> , <code>average</code> , <code>max</code> .

MVCC

MVCC отвечает за изоляцию транзакций. Он обнаруживает конфликты и гарантирует, что кортежи, которых больше нет в спейсе, но которые читаются какой-либо транзакцией (или могут быть прочитаны), не были удалены.

Метрика	Описание
<code>tnt_memtx_mvcc_trackers</code>	Трекеры, отслеживающие чтения транзакций. Всегда имеет метку <code>{kind="..."}</code> с возможными значениями <code>total</code> , <code>average</code> , <code>max</code> .
<code>tnt_memtx_mvcc_conflicts</code>	Выделяется в случае конфликтов транзакций. Всегда имеет метку <code>{kind="..."}</code> с возможными значениями <code>total</code> , <code>average</code> , <code>max</code> .

Кортежи (Tuples)

Сохраненные кортежи разделены на 3 категории: используемые (`used`), read view (`read_view`), отслеживающие (`tracking`). Для каждой категории есть две метрики: * `retained` - кортежи, которых больше нет в индексе, но MVCC не позволяет их удалить. * `stories` - механизм историй (story) в MVCC. Почти каждый кортеж имеет историю.

Метрика	Описание
<code>tnt_memtx_mvcc_tuples_used_stories</code>	Кортежи, используемые активными транзакциями чтения-записи. Метка <code>{kind="..."}</code> со значениями <code>count</code> (количество) и <code>total</code> (байты).
<code>tnt_memtx_mvcc_tuples_used_retained</code>	Удерживаемые кортежи, используемые активными транзакциями чтения-записи. Метка <code>{kind="..."}</code> .
<code>tnt_memtx_mvcc_tuples_read_view_stories</code>	Кортежи, не используемые активными транзакциями чтения-записи, но используемые транзакциями только для чтения (в read view). Метка <code>{kind="..."}</code> .
<code>tnt_memtx_mvcc_tuples_read_view_retained</code>	Удерживаемые кортежи, не используемые активными транзакциями чтения-записи, но используемые read-view транзакциями. Метка <code>{kind="..."}</code> .
<code>tnt_memtx_mvcc_tuples_tracking_stories</code>	Кортежи, не используемые напрямую никакими транзакциями, но используемые MVCC для отслеживания чтений. Метка <code>{kind="..."}</code> .
<code>tnt_memtx_mvcc_tuples_tracking_retained</code>	Удерживаемые кортежи, не используемые напрямую никакими транзакциями, но используемые MVCC для отслеживания чтений. Метка <code>{kind="..."}</code> .

Статистика Read View

Метрика	Описание
<code>tnt_memtx_tuples_data_total</code>	Общий объем памяти (в байтах), выделенный для кортежей данных. Включает значения метрик <code>tnt_memtx_tuples_data_read_view</code> и <code>tnt_memtx_tuples_data_garbage</code> , а также кортежи, фактически хранящиеся в спейсах memtx.
<code>tnt_memtx_tuples_data_read_view</code>	Память (в байтах), занятая под read view.

Метрика	Описание
<code>tnt_memtx_tuples_data_garbage</code>	Память (в байтах), которая не используется и запланирована к освобождению (освобождается лениво при выделении памяти).
<code>tnt_memtx_index_total</code>	Общий объем памяти (в байтах), выделенный для индексации данных. Включает значение метрики <code>tnt_memtx_index_read_view</code> плюс память, используемую для индексации кортежей, фактически хранящихся в спейсах memtx.
<code>tnt_memtx_index_read_view</code>	Память (в байтах), занятая под read view для индексов.

Конфигурация Tarantool

Начиная с версии: 3.0.0.

Метрика	Описание
<code>tnt_config_alerts</code>	Количество текущих предупреждений (<code>alerts</code>) при применении конфигурации инстанса. Метка <code>{level="warn"}</code> покрывает предупреждения, <code>{level="error"}</code> — ошибки.
<code>tnt_config_status</code>	Статус применения текущей конфигурации инстанса. Метка <code>status</code> содержит возможное имя статуса. Текущий статус имеет значение метрики 1, неактивные статусы — 0.

Пример набора метрик для статуса конфигурации:

 27 апреля 2026 г.