



Ай-ТИ Фреш

ТЕХНИЧЕСКИЙ РАЗБОР

Zabbix 7.4: счётчик DNS растёт, а Change per second — ноль

Numeric (float) и preprocessing для кумулятивных Windows-счётчиков DNS

Июль 2026

itfresh.ru · ИТ-аутсорсинг для юридических лиц

Суть проблемы

Типовой кейс: через perf_counter_en снимаем счётчик \DNS\Total Query Received, значение растёт, а шаг Change per second даёт ноль. Причина — Type of information = Numeric (unsigned): по документации 7.4 дробный результат принимается, но десятичная часть отбрасывается. При нагрузке ниже 1 запроса в секунду в истории всегда 0, график оживает только на нагрузочном тесте.

Почему это важно бизнесу

- Мониторинг показывает «тишину» на рабочем DNS — нет данных для планирования ёмкости и отчёта по SLA.
- Триггеры на rate()/change() не срабатывают при реальном росте запросов — аномалия видна только по отказу службы.
- Тренды по скорости запросов накапливаются нулями: прогноз роста нагрузки строить не на чем.
- Ошибка типа данных тиражируется на все счётчики-скорости хоста — диск, сеть, IIS, не только DNS.

Ключевые параметры реализации

7.4

Версия Zabbix, под которую закладываем шаблон DNS-счётчиков и шаги preprocessing
zabbix.com 7.4

float

Type of information для item с Change per second — на unsigned десятичная часть отбрасывается
Preprocessing 7.4

1/сек

Порог нагрузки, ниже которого Numeric (unsigned) всегда даёт 0 в Change per second
Preprocessing 7.4

1-900 с

Диапазон второго параметра perf_counter_en[counter,<interval>] — окно усреднения, дефолт 1
Windows keys 7.4

5 мин

Минимальное окно rate() в триггере по кумулятивному счётчику, чтобы не ловить дребезг
наш стандарт

60 с

Интервал опроса item по DNS-счётчику в нашем базовом шаблоне Windows-хоста
наш стандарт



Настройка item под DNS-счётчик через perf_counter_en

Что настраиваем

Windows Server с ролью DNS, Zabbix agent, счётчик \DNS\Total Query Received

Как мы это делаем

- 1 Item типа Zabbix agent, ключ perf_counter_en["\DNS\Total Query Received"], интервал опроса 60 с — счётчик кумулятивный, усреднение агентом не нужно.
- 2 Type of information = Numeric (float) выставляем до первого сбора: на unsigned десятичная часть результата отбрасывается.
- 3 Шаг preprocessing Change per second — на самом item либо на зависимом item, если raw-значение нужно хранить отдельно.
- 4 Проверяем через Test в форме item: подставляем prev/last и время, сравниваем результат на float и на unsigned.
- 5 Units = !qps: префикс «!» отключает автоматическое преобразование единиц, на тип хранения значения не влияет.

РЕЗУЛЬТАТ

Получаем реальный график скорости DNS-запросов даже при 1 запросе на 5–10 секунд: вместо нулевой линии видно фактический профиль нагрузки и суточные пики, пригодные для порогов и отчёта по ёмкости.

КЛЮЧЕВОЙ НЮАНС

Смена типа item на float задним числом не пересчитывает собранную историю — старые точки останутся усечёнными, сравнивать их с новыми напрямую нельзя.



Расчётный Change per second против нативного /sec-счётчика

Что настраиваем

Тот же DNS-хост: *Total Query Received* против *Total Query Received/sec*

Как мы это делаем

- 1 Снимаем нативный `perf_counter_en["\DNS\Total Query Received/sec"]` — Windows сам считает скорость и отдаёт дробное значение.
- 2 Параллельно считаем Change per second от *Total Query Received* на `float-item` и сверяем оба ряда на устойчивой нагрузке.
- 3 Основным оставляем расчётный вариант: единый паттерн `preprocessing` на все кумулятивные счётчики хоста.
- 4 Триггер строим на кумулятивном `item`: `rate(/host/dns.queries.raw,5m)>N` — `rate()` работает только с монотонно растущим счётчиком.
- 5 Альтернатива без `item-preprocessing` — `PerfCounterEn` в конфиге агента: имя, "путь счётчика", период усреднения.

РЕЗУЛЬТАТ

Единая методология `preprocessing` для DNS/IIS/диска/сети сокращает внедрение нового шаблона счётчика с часов до 15–20 минут на хост и снимает разноречивость в именах и типах `item`.

КЛЮЧЕВОЙ НЮАНС

Готовый `/sec`-счётчик Windows избавляет от шага `Change per second`, но не от требования `Numeric (float)`: дробное значение на `unsigned` всё равно усечётся.



Хранение и триггеры: history/trends на float-item

Что настраиваем

*Item dns.query.rate: History 90d, Trends 730d, raw-item
dns.queries.raw рядом*

Как мы это делаем

- 1 History Storage Period 90d, Trend Storage Period 730d — trends пишут avg/min/max, на float без округления до целого.
- 2 В триггерах разводим функции по сигнатурам 7.4:
change(/host/dns.query.rate) без периода и
rate(/host/dns.queries.raw,10m) с окном.
- 3 Пороги подбираем по собранной истории — Latest data и графики за 7–30 дней, выражение собираем через конструктор.
- 4 Отдельно проверяем участок смены типа item: на границе unsigned→float пороги считаем только по новым точкам.

РЕЗУЛЬТАТ

Клиент получает отчёт по capacity planning DNS без искажений от усечения: тренды пригодны для прогноза роста нагрузки на 6–12 месяцев и для обоснования апгрейда.

КЛЮЧЕВОЙ НЮАНС

Если item раньше был unsigned, старые trends остаются округлёнными и не сравнимы с новыми float-данными — точку перехода фиксируем в описании item.



Подводные камни

✗ **Numeric (unsigned) на скоростном item**

Change per second считает дробь, но на unsigned десятичная часть отбрасывается — при низкой нагрузке в истории всегда 0.

✗ **Смена типа после сбора истории**

Zabbix не пересчитывает старые записи при переходе unsigned→float — на границе смены типа график и тренды несравнимы.

✗ **Custom multiplier поверх unsigned**

На unsigned дробная часть входящего значения отбрасывается до применения множителя, результат тоже усекается — 0.1 даёт ноль.

✗ **Нет baseline у Change per second**

Первое значение после создания item или перезапуска сервера отбрасывается: предыдущего значения нет, график стартует на интервал позже.

✗ **Числовой индекс счётчика вместо имени**

Индексы Perflib различаются между машинами и версиями Windows и нечитаемы в шаблоне — берём perf_counter_en с именем из Perflib\009.

✗ **Смещение raw- и /сек-счётчика**

Оба item без пометки в имени — на разборе инцидента путаем, где прошёл Change per second, а где счётчик уже отдал готовую скорость.

✗ **rate() поверх item со скоростью**

rate() определён только для монотонно растущего счётчика: на уже посчитанной скорости результат бессмысленный, берём raw-item.

✗ **Крупный интервал опроса при низкой нагрузке**

Интервал 300 с при малом трафике даёт грубые дискретные шаги Change per second — триггер реагирует с большим опозданием.

Как правильно

МИНИМУМ

- Item на perf_counter_en, Type of information = Numeric (float), интервал 60 с
- Один шаг preprocessing Change per second на кумулятивном счётчике DNS
- Test в форме item перед сохранением — сверяем результат шага на реальных значениях

НОРМАЛЬНО

- Отдельный float-item под скорость, raw-item хранится параллельно для rate()
- Триггеры: rate()/avg() с окном от 5 минут, change() — без параметра периода
- Units = !qps на item: читаемый график без автоконвертации единиц
- Регламент: переход unsigned→float только на новом item, без миграции истории

ХОРОШО

- Единый шаблон preprocessing для всех кумулятивных Windows-счётчиков хоста
- Trend Storage Period не менее 365 дней на float-items для capacity planning
- Сверка расчётного Change per second с нативным /sec-счётчиком при внедрении
- Чек-лист типов данных перед публикацией шаблона в репозиторий клиента



Чек-лист самопроверки

- | | |
|--|--|
| ☐ Type of information на item со скоростью запросов установлен Numeric (float)? | ☐ Окно rate()/avg() в триггере не короче 5 минут, а change() задан без параметра периода? |
| ☐ Результат шага Change per second проверен через Test в форме item на реальных значениях? | ☐ Trend Storage Period на float-item достаточен для отчёта по capacity planning? |
| ☐ Используется perf_counter_en с именем счётчика, а не числовой индекс Perflib? | ☐ Задан Units (!qps) для читаемости графика без автоконвертации единиц? |
| ☐ Есть отдельный raw-item для кумулятивного счётчика, независимый от расчётного float-item? | ☐ Учтено, что смена типа item не пересчитывает уже собранную историю и тренды? |
| ☐ rate() в триггере применяется к кумулятивному item, а не к уже посчитанной скорости? | ☐ Custom multiplier (если используется) применяется на item с типом float, а не unsigned? |

Если хотя бы на два вопроса ответ «нет» или «не знаю» — тема требует внимания.



Как поможет ITFresh

ITFresh — ИТ-аутсорсинг для юридических лиц до 50 рабочих мест в Москве и области. 15+ лет практики, собственная инфраструктура в дата-центре МТС (8 серверов Dell Xeon Platinum).

- Настраиваем шаблоны Zabbix 7.4 под Windows-счётчики DNS/IIS/диска с float и preprocessing.
- Аудируем шаблоны клиента на усечение Change per second и переводим item на Numeric (float).
- Строим триггеры rate()/avg()/change() под реальный профиль нагрузки, без дребезга и тишины.
- Проводим нагрузочный тест DNS и валидируем графики скорости до передачи в эксплуатацию.
- Ведём чек-листы типов данных Zabbix-шаблонов для инфраструктур клиентов до 50 РМ.

15+

лет в ИТ-поддержке

50

рабочих мест — наш профиль

МТС

дата-центр, Москва



КОНТАКТЫ

Обсудить вашу задачу

Сайт **itfresh.ru**

Телефон **+7 903 729-62-41**

Telegram **@ITfresh_Boss**

Бесплатно посмотрим вашу инфраструктуру по этому чек-листу и скажем, где тонко — без обязательств.



itfresh.ru



Техническая база

- 01** Item value preprocessing — Change per second, Custom multiplier (zabbix.com — 7.4)
- 02** Creating an item — Type of information, Units (zabbix.com — 7.4)
- 03** Windows-specific item keys — perf_counter_en[counter,<interval>] (zabbix.com — 7.4)
- 04** History functions — change(), rate() (zabbix.com — 7.4)
- 05** Configuration parameters: Zabbix agent (Windows) — PerfCounterEn (zabbix.com — 7.4)
- 06** DNS Object — счётчики Total Query Received и .../sec (learn.microsoft.com — 2025)

