Engineer

Метрика

число, яке система повідомляє про власний стан

Що це таке

Метрика — це вимірювання працюючого сервісу, зроблене в певний момент і збережене як значення з позначкою часу та кількома мітками, щоб ту саму величину можна було прочитати знову через хвилину, добу чи місяць. Одна названа величина — це метрика; збережена як послідовність значень у часі, вона стає часовим рядом; а OpenTelemetry відносить метрики до своїх сигналів, поряд із трасами й журналами. Обслужені запити, зайнята пам'ять і завдання, що чекають у черзі, — типові метрики.

Від інших сигналів метрики відрізняє агрегація. Журнал зберігає один рядок на подію, а траса стежить за одним запитом крізь систему. Метрика згортає багато вимірювань у кілька статистик за часове вікно: кількість, суму, поточне значення, розподіл за кошиками. Документація OpenTelemetry називає результат статистичною інформацією в сукупності, і в цьому і сила, і межа: метрика каже, скільки запитів завершилися помилкою за останню хвилину, а журнал — які саме.

OpenTelemetry і Prometheus сходяться на невеликому наборі форм, і вибір правильної — це більша частина проєктування:

- **Лічильник** (counter) накопичує і лише зростає, як одометр в автомобілі: обслужені запити, надіслані байти, помічені помилки. Prometheus допускає один виняток — скидання до нуля, коли процес перезапускається.
- **Двонапрямний лічильник** (up-down counter), у термінах OpenTelemetry, накопичує, але може й зменшуватися, як довжина черги, коли робота надходить і йде.
- **Датчик** (gauge) — поточне значення, прочитане в момент читання, як покажчик пального в тому ж автомобілі: зайнята пам'ять, температура, запити в обробці.

- **Гістограма** розкладає вимірювання за кошиками і зберігає їхню кількість і суму, тож може відповісти, скільки запитів тривали менше за секунду.

Prometheus має ще й тип summary, що обчислює квантилі всередині програми. Модель даних OpenTelemetry зберігає його лише як застарілий тип, бо summary не завжди можна змістовно об’єднати.

Текстовий формат, який збирає Prometheus, показує всі частини разом: рядок довідки, рядок типу, а далі по рядку на кожну комбінацію міток з іменем, мітками, значенням і необов’язковою позначкою часу в мілісекундах від епохи.

```
# HELP http_requests_total The total number of HTTP requests.
# TYPE http_requests_total counter
http_requests_total{method="post",code="200"} 1027 1395066363000
http_requests_total{method="post",code="400"} 3 1395066363000
```

Застосуйте поняття до рішення

Застосуйте «Що таке метрика?» до конкретного рішення. Ми можемо пов'язати поняття із системою, яку ви створюєте або купуєте. Моніторинг і сповіщення, які попереджають, поки збій ще малий, і цифри, що показують, як часто так і залишається.

Переглянути наші послуги Зв'язатися з нами →

Перевірте припущення за терміном

Використайте пояснення та пов'язані роботи, щоб сформулювати запитання для проєкту. Ми можемо перевірити припущення разом до вибору інструмента. Запустила перший дашборд спостережуваності — перший огляд власних систем компанії

Запустила перший дашборд спостережуваності Зв'язатися з нами →

Чому це важливо

Метрики досить дешеві, щоб тримати їх для всього. Пам’ять, якої вони потребують, зростає з кількістю різних комбінацій міток, а не з трафіком; документація OpenTelemetry каже про це прямо, протиставляючи метрики журналам. Лічильник, збільшений мільярд разів, — це все одно одне число. Саме це дає команді змогу рахувати кожен запит, а не вибірку, і тримати підрахунки достатньо довго, щоб порівняти цей місяць із попереднім.

Вони перетворюють суперечки на вимірювання. Чи достатньо великий хост, чи сповільнив реліз відповіді, як скоро заповниться диск: на кожне питання є відповідь у ряді, записаному ще до того, як хтось запитав. На тих самих

числах будують сповіщення, бо правило сповіщення — це умова над метрикою, через яку, як хтось вирішив, варто прокинутися.

Вони також переживають код, що їх видає, тож форму варто зробити правильною. Порадник Prometheus з іменування просить одну одиницю й одну величину на метрику, у базових одиницях, як-от секунди й байти, а не мілісекунди чи мегабайти, з одиницею як суфіксом у множині й `_total` для накопичуваної кількості. Тоді імена на кшталт `http_request_duration_seconds` і `process_cpu_seconds_total` самі кажуть, що в них.

Важить і те, як мандрують підсумки. Prometheus передає кумулятивні підсумки, тож прогалини від збору, що час від часу збоїть, усереднюються, а ціна — пам'ять у відправника пропорційно до його комбінацій міток. StatsD передає дельти, тобто зміну від останнього звіту, і це виносить ці витрати за межі процесу.

Як це виглядає на практиці

Власний хост публічних сервісів компанії — одноядерна машина з 512 МБ пам'яті, з яких придатні близько 464 МБ, і на ній пам'ять стала аргументом, що визначав рішення. Звітність там працює на запит: сім команд, які зчитують стан хоста і формують Markdown, нічого не змінюють та виконуються лише на запит. Коли постало питання, чи витримає цей один хост із 512 МБ ще й onion-сервіс, відповідь спиралася не на думку, а на вибірки вимірювань, зафіксовані протягом доби: доступна пам'ять не опускалася нижче приблизно 310 МБ із 464, використання підкачування трималося на 2,6%, а процесор 99,7% часу простоював.

Згодом ці звіти мали сказати, чого варті їхні числа. Шість збирачів лише для читання тепер кажуть, чого вони не прочитали, відколи стало ясно, що сліпий збір і здоровий хост дають ту саму сторінку: дзеркало, чий демон змінив формат журналу, давало той самий нуль, що й дзеркало, якого ніхто не відвідував. Пороги підігнали до 116 зразків, які звіт метрик уже виміряв, а поряд із межами записали спостережені діапазони, і звіт розрізняє три наслідки: порушення, відсутність порушення і неоцінено, бо ніщо цього не виміряло. Кожен збирач повідомляє свій час роботи і свою сліпоту, як-от кількість прочитаних файлів і файлів, які не вдалося прочитати, навіть коли відповідь нуль. Межі застосовуються до середніх добових значень, які можуть приховати однохвилинний сплеск. Перемикач, що запускається

кожні п’ятнадцять хвилин, під час кожного запуску перевіряє доступну пам’ять і вільне місце на диску, але короткочасне порушення межі між перевірками все одно може залишитися непоміченим.

Найкорисніші метрики часто походять від задачі, а не від хоста. Нотатка про продаж обмеженого запасу без продажів понад залишок збирає на одній панелі очікування блокувань, резерви, що спливли, і розрив між числом шлюзу та числом бази, а інваріант `available + held + sold = total` запускає за розкладом, з графіком і тривогою на будь-яке розходження. Кожне з цих чисел походить зі знання того, як саме цей продаж іде не так.

Де виникають проблеми

Метрики рідко ламаються гучно. Графік далі малюється, а число на ньому тихо означає щось інше, ніж каже його назва.

Значення, що спадає, публікують як лічильник. Сторінка Prometheus про типи метрик радить не використовувати лічильник для того, що може зменшуватися, як-от кількість запущених процесів: це датчик. Функції запитів сприймають будь-яке падіння лічильника як перезапуск і виправляють його, тож значення, що спадає навмисно, з’являється як зростання, якого ніколи не було.

Одиниці розходяться між сервісами. Один сервіс звітує тривалість у мілісекундах, інший — у секундах, і панель, що їх порівнює, помиляється в тисячу разів. Порадник з іменування каже, що метрика мусить стосуватися однієї одиниці й має використовувати базові одиниці. Перевести один раз, біля джерела, дешевше, ніж пам’ятати різницю в кожному запиті.

Сума за мітками нічого не означає. Емпіричне правило порадника з іменування: сума чи середнє за всіма мітками метрики має бути змістовним. Його власний приклад: місткості кількох черг належать до однієї метрики, а місткість черги та її поточна довжина — ні.

Невиміряне читається як нуль. Збирач, що перестав читати свій вхід, лишає нуль або відсутній рядок, що виглядає точнісінько як спокійна система. Модель даних OpenTelemetry має прапорець для точки без записаного

значення, свій відповідник маркера застарілості Prometheus, бо відсутні дані й нульове значення — різні факти. Звітуйте їх по-різному й дайте «неоцінено» окремий наслідок, як це роблять збирачі вище.

Вікно ховає сплеск. Датчик — це зразок у певний момент, і модель даних OpenTelemetry передає лише останнє значення в кожному інтервалі, тож сплеск між двома зразками не лишає сліду. Денне середнє так само згладжує хвилину біля стелі. Обирайте вікно з огляду на збій, який треба побачити, і тримайте швидку перевірку поряд із повільною, коли важливі обидві.

Переповнення занижує відфільтрований запит. SDK OpenTelemetry за замовчуванням відстежує щонайбільше 2000 різних комбінацій атрибутів на потік метрики, а всі наступні згортає в одну точку даних з позначкою `otel.metric.overflow=true`. Загальний підсумок лишається правильним, але переповнення замінює всю комбінацію, а не лише її шумну частину. У власному прикладі документації лічильник запитів несе `url.path` і `success`; коли він переповнюється, запит на `success=false` дає занижену кількість, і сповіщення про частку помилок, побудоване на ньому, може перестати спрацьовувати. Тримайте атрибути в малих, відомих наборах і стежте за позначкою переповнення.

Що почитати далі

- OpenTelemetry: Metrics — види інструментів, агрегація та обмеження кардинальності.
- OpenTelemetry: Metrics data model — датчики, суми, гістограми, темпоральність і прапорець відсутнього значення.
- Prometheus: Metric types — counter, gauge, histogram і summary.
- Prometheus: Metric and label naming — базові одиниці, суфікси одиниць і `_total`.
- Prometheus: Histograms and summaries — де обчислюються квантилі і які з них можна поєднувати.
- Prometheus: Exposition formats — текстовий формат рядок за рядком.

Словник

Обирайте за фактичними умовами

Опишіть шлях розгортання, експлуатаційні ризики та чергову команду. Ми можемо визначити зміну з перевірками, шляхом відновлення та чіткою відповідальністю.

Дивитися, скільки це коштує Зв'язатися з нами →

Зробіть термінологію практичною

Запитайте, як «Що таке метрика?» стосується вашої організації. Конкретного прикладу з поточної системи достатньо для початку розмови.

Переглянути наші послуги

welcome@engineer.company +45 50 10 61 84 Копенгаген, Данія

Візьміть цю сторінку із собою

Візитівка

Дбайливо зроблено з любов'ю — D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Авторське право © 2025 – дотепер · Engineer ApS · Усі права захищено.

Надруковано з https://engineer.company/uk/glossary/metrics/