Engineer

Metrikker

tall et system rapporterer om seg selv

Hva det er

Metrikker er målinger av en tjeneste i drift. Hver måling tas på et bestemt tidspunkt og lagres som en verdi med et tidsstempel og noen få etiketter (labels), slik at den samme størrelsen kan leses igjen et minutt, en dag eller en måned senere. Én navngitt størrelse er en metrikk, og lagret som en rekke verdier over tid blir den en tidsserie. OpenTelemetry regner metrikkene som ett av sine signaler, ved siden av sporinger (traces) og logger. Antall besvarte forespørsler, minne i bruk og jobber som venter i en kø er typiske eksempler.

Det som skiller metrikker fra de andre signalene, er aggregering. En logg har én linje per hendelse, og en sporing følger én forespørsel gjennom systemet. En metrikk slår mange målinger sammen til noen få tall over et tidsvindu: et antall, en sum, en nåverdi eller en fordeling på bøtter. OpenTelemetry-dokumentasjonen kaller resultatet aggregert statistisk informasjon, og det er både styrken og begrensningen: en metrikk sier hvor mange forespørsler som feilet det siste minuttet, mens en logg sier hvilke.

OpenTelemetry og Prometheus er enige om et lite sett med former, og å velge riktig form er det meste av designarbeidet:

- En **teller** (counter) akkumulerer og går bare oppover, som kilometertelleren i en bil: besvarte forespørsler, sendte byte, feil som er registrert. Prometheus tillater ett unntak, en nullstilling når prosessen starter på nytt.
- En **opp-ned-teller** (up-down counter), som OpenTelemetry kaller det, akkumulerer, men kan også gå ned, f.eks. lengden på en kø etter hvert som arbeid kommer inn og går ut.
- En **måler** (gauge) er en nåverdi, lest i det øyeblikket den leses, som drivstoffmåleren i den samme bilen: minne i bruk, en temperatur, forespørsler under behandling.

- Et **histogram** sorterer målinger i bøtter og holder på antall og sum, så det kan svare på hvor mange forespørsler som tok under ett sekund.

Prometheus har i tillegg en summary, som regner ut kvantiler inne i programmet. OpenTelemetrys datamodell beholder den bare som en eldre type, fordi summaries ikke alltid lar seg slå sammen på en meningsfull måte.

Tekstformatet Prometheus henter (scraper), viser bitene samlet: en hjelpelinje, en typelinje og deretter én linje per kombinasjon av etiketter, med navnet, etikettene, verdien og et valgfritt tidsstempel i millisekunder siden epoken.

```
# 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
```

Bruk begrepet i en beslutning

Bruk «Hva er metrikker?» i en konkret beslutning. Vi kan knytte begrepet til systemet dere bygger eller kjøper. Overvåking og varsler som sier fra mens en feil fortsatt er liten, og tallene som viser hvor ofte den holder seg liten.

Se tjenestene våre Ta kontakt →

Undersøk antakelsene bak ordet

Bruk forklaringen og det tilknyttede arbeidet til å formulere spørsmål for prosjektet. Vi kan undersøke antakelsene sammen før et verktøy velges. Bygde selskapets første observerbarhetsdashbord — selskapets første innsyn i egne systemer

Bygde selskapets første observerbarhetsdashbord Ta kontakt →

Hvorfor det er viktig

Metrikker er billige nok til at du kan ha dem for alt. Minnet de krever, vokser med antall ulike kombinasjoner av etiketter, ikke med trafikken, og OpenTelemetry-dokumentasjonen sier det rett ut og stiller det opp mot logger. En teller som er talt opp en milliard ganger, er fortsatt ett tall. Det er det som lar et team telle hver eneste forespørsel i stedet for et utvalg, og beholde tallene lenge nok til å sammenligne denne måneden med forrige.

De gjør diskusjoner om til målinger. Er verten stor nok, gjorde den siste utgivelsen svartiden lengre, hvor fort blir disken full? Hvert av spørsmålene har et svar i en tidsserie som ble registrert før noen spurte. Et driftsvarsel bygges på de samme tallene, for regelen bak det er et vilkår på en metrikk som noen mente var verdt å bli vekket for.

Metrikkene lever også lenger enn koden som sender dem ut, så formen er verdt å få riktig. Navneveiledningen til Prometheus ber om én enhet og én størrelse per metrikk, i grunnenheter som sekunder og byte i stedet for millisekunder eller megabyte, med enheten som suffiks i flertall og `_total` på et akkumulerende antall. Navn som `http_request_duration_seconds` og `process_cpu_seconds_total` sier da selv hva de inneholder.

Hvordan totalene reiser, betyr også noe. Prometheus rapporterer kumulative totaler, så hull etter en innsamling som feiler av og til, jevner seg ut, mot at avsenderen bruker minne i takt med antall kombinasjoner av etiketter. StatsD rapporterer deltaer, altså endringen siden forrige rapport, og flytter dermed den kostnaden ut av prosessen.

Slik ser det ut i praksis

Selskapets egen vert for offentlig tilgjengelige tjenester er en maskin med én kjerne og 512 MB minne, hvorav omtrent 464 MB er brukbart. Der ble minnet argumentet som avgjorde beslutningene. Rapporteringen hentes på forespørsel: sju kommandoer som leser verten og gjengir Markdown, ikke endrer noe og bare kjører når noen ber om det. Da spørsmålet kom opp om denne ene verten med 512 MB også kunne romme en onion-tjeneste, kom svaret fra et dokumentert døgn med stikkprøvemålinger i stedet for en mening: tilgjengelig minne falt aldri under omtrent 310 MB av 464, swap-bruken lå på 2,6 %, og prosessoren var ledig 99,7 % av tiden.

Senere måtte de samme rapportene si hva tallene deres var verdt. Seks innsamlere som bare leser ble pålagt å oppgi det de aldri leste da det ble klart at en blind innsamling og en vert der alt er i orden, ga den samme siden. Et speil der daemonen hadde endret loggformat, ga den samme nullen som et speil ingen besøkte. Tersklene ble tilpasset de 116 målepunktene som metrikkrapporten allerede hadde målt, med de observerte spennene registrert ved siden av grensene, og rapporten skiller mellom tre utfall: et brudd, ikke noe brudd, og ikke vurdert fordi ingenting målte det. Hver innsamler oppgir kjøretiden og blindsonene sine, f.eks. filer lest mot filer som ikke kunne leses, også når svaret er null. Grensene gjelder døgngjennomsnitt, som kan skjule en topp som varer i ett minutt. Dødmannsknappen som kjører hvert femtende minutt, sjekker tilgjengelig minne og ledig diskplass ved hver kjøring, men et kortvarig grensebrudd mellom kontrollene kan likevel forbli uoppdaget.

De nyttigste metrikkene kommer ofte fra problemet og ikke fra verten. Notatet om oversalg i et lynsalg samler låseventing, utløpte reservasjoner og avstanden mellom Redis-telleren foran salget og tallet i databasen på ett dashbord. Det kjører også invarianten `available + held + sold = total` etter en tidsplan, viser den i en graf og sender varsel ved ethvert avvik. Hvert av de tallene kommer av å vite hvordan akkurat det salget går galt.

Hvor det går galt

Metrikker går sjelden i stykker med et smell. Grafen fortsetter å tegne, og tallet på den betyr noe annet enn det navnet sier, uten at noen får beskjed.

En verdi som synker, publiseres som teller. Prometheus-siden om metrikktyper sier at en teller ikke skal brukes til noe som kan gå ned, f.eks. antall kjørende prosesser: det er en måler. Spørringsfunksjonene tolker ethvert fall i en teller som en omstart og korrigerer for det, så en verdi som synker med vilje, dukker opp som økninger som aldri skjedde.

To tjenester måler tid i hver sin enhet. Én tjeneste rapporterer en varighet i millisekunder, en annen i sekunder, og et panel som sammenligner dem, bommer med en faktor på tusen. Navneveiledningen sier at en metrikk må gjelde én enkelt enhet og bør bruke grunnenheter. Å regne om én gang, ved kilden, er billigere enn å huske forskjellen i hver eneste spørring.

Summen over alle etikettene er meningsløs. Tommelfingerregelen i navneveiledningen er at summen eller gjennomsnittet over alle etikettene til en metrikk skal gi mening. Veiledningens eget eksempel: kapasiteten til flere køer hører sammen i én metrikk, mens kapasiteten til en kø og den nåværende lengden på den samme køen ikke gjør det.

En blind innsamler ser ut som et rolig system. En innsamler som har sluttet å lese inndataene sine, etterlater en null eller en manglende rad, og begge ser nøyaktig ut som et system der ingenting skjer. OpenTelemetrys datamodell har et flagg for et datapunkt uten registrert verdi, motstykket til staleness-markøren i Prometheus, fordi manglende data og verdien null er to forskjellige fakta. Rapporter dem forskjellig, og gi «ikke vurdert» et eget utfall, slik innsamlerne over gjør.

Toppen faller mellom to målinger. En måler er en stikkprøve i ett øyeblikk, og OpenTelemetrys datamodell rapporterer bare den siste verdien i hvert intervall, så et utbrudd mellom to stikkprøver etterlater ingen spor. Et døgngjennomsnitt glatter ut et minutt ved taket på samme måte. Velg vinduet ut fra den feilen som må bli synlig, og hold en rask kontroll ved siden av en langsom når begge betyr noe.

Feilene forsvinner inn i en overløpsrad. OpenTelemetry-SDK-en holder som standard styr på høyst 2 000 ulike kombinasjoner av attributter per metrikkstrøm og legger alle nye inn i ett datapunkt merket `otel.metric.overflow=true`. Totalen blir riktig, men overløpet erstatter hele kombinasjonen og ikke bare den støyende delen av den. I dokumentasjonens eget eksempel har en teller for forespørsler attributtene `url.path` og `success`. Når den flyter over, teller en spørring på `success=false` for lavt, og et driftsvarsel for feilraten som bygger på den, kan slutte å utløses. Hold attributtene til små, kjente mengder, og følg med på overløpsmarkøren.

Les mer

- OpenTelemetry: Metrics — typene instrumenter, aggregeringen og grensen for antall kombinasjoner.
- OpenTelemetry: Metrics data model — målere, summer, histogrammer, temporalitet og flagget for manglende verdi.
- Prometheus: Metric types — teller, måler, histogram og summary.
- Prometheus: Metric and label naming — grunnenheter, enhetssuffikser og `_total`.
- Prometheus: Histograms and summaries — hvor kvantilene regnes ut, og hvilke som kan slås sammen.
- Prometheus: Exposition formats — tekstformatet linje for linje.

Ordliste

Velg ut fra deres faktiske rammer

Ta med utrullingsløpet, driftsrisikoene og dem som har vakt. Vi kan avgrense en endring med kontroller, en vei tilbake og tydelig ansvar.

Se hva det koster Ta kontakt →

Gjør begrepene praktiske

Spør hvordan «Hva er metrikker?» gjelder i organisasjonen deres. Et konkret eksempel fra dagens system er nok til å starte samtalen.

Se tjenestene våre

welcome@engineer.company +45 50 10 61 84 København, Danmark

Ta med deg denne siden

Visittkort

Laget med kjærlighet av D.

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

Opphavsrett © 2025 – nå · Engineer ApS · Alle rettigheter forbeholdt.

Skrevet ut fra https://engineer.company/nb/glossary/metrics/