Engineer

Metrikker

tal, et system rapporterer om sig selv

Hvad det er

Metrikker er målinger af en kørende tjeneste, hver taget på et bestemt tidspunkt og gemt som en værdi med et tidsstempel og nogle få etiketter, så den samme størrelse kan aflæses igen et minut, en dag eller en måned senere. Én navngiven størrelse er en metrik; gemt som en række værdier over tid bliver den til en tidsserie; og OpenTelemetry regner metrikkerne som et af sine signaler ved siden af traces og logs. Betjente forespørgsler, brugt hukommelse og job, der venter i en kø, er typiske metrikker.

Det, der skiller metrikker fra de andre signaler, er aggregering. En log gemmer én linje pr. hændelse, og en trace følger én forespørgsel gennem systemet. En metrik folder mange målinger sammen til nogle få statistikker over et tidsvindue: et antal, en sum, en aktuel værdi, en fordeling over spande. OpenTelemetry-dokumentationen kalder resultatet statistisk information i samlet form, og det er både styrken og grænsen: en metrik siger, hvor mange forespørgsler der fejlede det seneste minut, og en log siger hvilke.

OpenTelemetry og Prometheus er enige om et lille sæt former, og at vælge den rigtige er det meste af designet:

- En **tæller** (counter) samler op og går kun op, som kilometertælleren i en bil: betjente forespørgsler, sendte bytes, sete fejl. Prometheus tillader én undtagelse, en nulstilling, når processen genstarter.
- En **op-ned-tæller** (up-down counter), i OpenTelemetrys sprog, samler op, men kan også gå ned, som længden af en kø, mens arbejde kommer til og går.
- En **måler** (gauge) er en aktuel værdi, aflæst i det øjeblik den aflæses, som benzinmåleren i den samme bil: brugt hukommelse, en temperatur, forespørgsler under behandling.
- Et **histogram** sorterer målinger i spande og gemmer deres antal og sum, så det kan svare på, hvor mange forespørgsler der tog under et sekund.

Prometheus har også typen summary, som beregner kvantiler inde i programmet. OpenTelemetrys datamodel beholder den kun som en ældre type, fordi summaries ikke altid lader sig slå sammen på en meningsfuld måde.

Tekstformatet, som Prometheus henter, viser delene samlet: en hjælpelinje, en typelinje og derefter én linje pr. kombination af etiketter med navnet, etiketterne, værdien og et valgfrit 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
```

Brug begrebet i en beslutning

Brug »Hvad er metrikker?« i en konkret beslutning. Vi kan knytte begrebet til det system, I bygger eller køber. Overvågning og alarmer, der siger til, mens fejlen stadig er lille, og tallene, der viser, hvor tit den bliver ved med at være det.

Se vores ydelser Kontakt os →

Undersøg antagelserne bag ordet

Brug forklaringen og det tilknyttede arbejde til at formulere spørgsmål til jeres projekt. Vi kan undersøge antagelserne sammen, før et værktøj vælges. Har bygget det første observerbarhedsdashboard — virksomhedens første overblik over sine egne systemer

Har bygget det første observerbarhedsdashboard Kontakt os →

Hvorfor det betyder noget

Metrikker er billige nok til at blive gemt for alt. Den hukommelse, de kræver, vokser med antallet af forskellige kombinationer af etiketter, ikke med trafikken; OpenTelemetry-dokumentationen siger det ligeud og stiller det op mod logs. En tæller, der er talt op en milliard gange, er stadig ét tal. Det er det, der lader et team tælle hver forespørgsel frem for et udsnit af dem og gemme tallene længe nok til at sammenligne denne måned med den forrige.

De gør diskussioner til målinger. Om en vært er stor nok, om en udgivelse gjorde svarene langsommere, hvor hurtigt en disk fyldes op: hvert spørgsmål har et svar i en serie, der blev optaget, før nogen spurgte. Alarmer bygges på de samme tal, for en alarmregel er en betingelse over en metrik, som nogen har besluttet er værd at blive vækket for.

De lever også længere end den kode, der udsender dem, så formen er værd at få rigtig. Prometheus' vejledning om navngivning beder om én enhed og én størrelse pr. metrik, i grundenheder som sekunder og bytes frem for millisekunder eller megabytes, med enheden som et suffiks i flertal og `_total` på et opsamlet antal. Navne som `http_request_duration_seconds` og `process_cpu_seconds_total` siger så direkte, hvad de rummer.

Hvordan totalerne rejser, betyder også noget. Prometheus rapporterer kumulative totaler, så huller fra en indsamling, der svigter en gang imellem, udjævnes, mod en pris i hukommelse hos afsenderen i forhold til dens kombinationer af etiketter. StatsD rapporterer deltaer, ændringen siden sidste rapport, hvilket flytter den udgift ud af processen.

Sådan ser det ud i praksis

Virksomhedens egen vært til offentligt tilgængelige tjenester har én kerne og 512 MB hukommelse, hvoraf omkring 464 MB kan bruges, og på den blev hukommelsen det argument, der afgjorde beslutninger. Rapportering er dér et pull: syv kommandoer, der læser værten og gengiver Markdown, intet ændrer og kun kører, når nogen beder om det. Da spørgsmålet kom op, om den ene vært på 512 MB også kunne bære en onion-tjeneste, kom svaret fra et registreret døgns målte stikprøver frem for fra en holdning: den laveste målte tilgængelige hukommelse var omkring 310 MB af 464, swap lå på 2,6 procent, og processoren var inaktiv 99,7 procent af tiden.

De rapporter skulle senere sige, hvad deres tal var værd. Seks indsamlere, der kun læser fik til opgave at oplyse, hvad de aldrig læste, da det stod klart, at en blind indsamling og en sund vært gav den samme side: et spejl, hvis dæmon ændrede sit logformat, gav det samme nul som et spejl, ingen besøgte. Tærsklerne blev tilpasset de 116 stikprøver, metrikrapporten allerede havde målt, med de observerede intervaller skrevet ved siden af grænserne, og rapporten skelner mellem tre udfald: et brud, intet brud, og ikke bedømt, fordi intet målte det. Hver indsamler rapporterer sin køretid og sin blindhed, som læste filer mod ulæselige filer, også når svaret er nul. Grænserne gælder for daglige gennemsnit, som kan skjule en spidsværdi, der varer ét minut. Kontakten, der kører hvert kvarter, kontrollerer den tilgængelige hukommelse og den ledige diskplads ved hver kørsel, men et kortvarigt brud på en grænse mellem kontrollerne kan stadig gå ubemærket hen.

De mest nyttige metrikker kommer ofte fra problemet frem for fra værten. Noten om at sælge et begrænset lager uden at oversælge samler låseventetider, udløbne reservationer og afstanden mellem portens tal og databasens på ét dashboard og kører invarianten `available + held + sold = total` efter en tidsplan, tegnet som graf og med alarm ved enhver afvigelse. Hvert af de tal kommer af at vide, hvordan netop det salg går galt.

Hvor det går galt

Metrikker går sjældent i stykker med et brag. Grafen bliver ved med at tegne, og tallet på den betyder stille og roligt noget andet end det, navnet siger.

En værdi, der falder, udgives som en tæller. Prometheus' side om metriktyper siger, at man ikke skal bruge en tæller til noget, der kan falde, som antallet af kørende processer: det er en måler. Forespørgselsfunktionerne behandler ethvert fald i en tæller som en genstart og retter for det, så en værdi, der falder med vilje, viser sig som stigninger, der aldrig skete.

Enhederne glider mellem tjenester. Én tjeneste rapporterer en varighed i millisekunder, en anden i sekunder, og et panel, der sammenligner dem, rammer ved siden af med en faktor tusind. Navngivningsvejledningen siger, at en metrik skal henvise til én enhed og bør bruge grundenheder. At omregne én gang, ved kilden, er billigere end at huske forskellen i hver forespørgsel.

En sum på tværs af etiketter betyder ingenting. Navngivningsvejledningens tommelfingerregel er, at summen eller gennemsnittet over alle en metriks etiketter skal give mening. Dens eget eksempel: kapaciteten af flere køer hører sammen i én metrik, mens en køs kapacitet og dens aktuelle længde ikke gør.

Det umålte læses som nul. En indsamler, der er holdt op med at læse sit input, efterlader et nul eller en manglende række, der ligner et stille system på en prik. OpenTelemetrys datamodel har et flag for et datapunkt uden registreret værdi, dens modstykke til Prometheus' staleness-markør, fordi manglende data og værdien nul er to forskellige fakta. Rapportér dem forskelligt, og giv "ikke bedømt" sit eget udfald, som indsamlerne ovenfor gør.

Vinduet skjuler toppen. En måler er en stikprøve i et øjeblik, og OpenTelemetrys datamodel rapporterer kun den sidste værdi i hvert interval, så et udbrud mellem to stikprøver efterlader intet spor. Et dagligt gennemsnit udglatter et minut ved

loftet på samme måde. Vælg vinduet ud fra den fejl, der skal ses, og hold en hurtig kontrol ved siden af en langsom, når begge betyder noget.

Overløb tæller for lavt i en filtreret forespørgsel. OpenTelemetrys SDK holder som standard styr på højst 2000 forskellige kombinationer af attributter pr. metrikstrøm og folder alle yderligere ind i ét datapunkt markeret `otel.metric.overflow=true`. Totalen forbliver korrekt, men overløbet erstatter hele kombinationen, ikke kun dens støjende del. I dokumentationens eget eksempel bærer en tæller af forespørgsler `url.path` og `success`; når den løber over, tæller en forespørgsel på `success=false` for lavt, og en alarm for fejlraten bygget på den kan holde op med at gå af. Hold attributterne til små, kendte mængder, og hold øje med overløbsmarkøren.

Læs mere

- OpenTelemetry: Metrics — instrumenttyper, aggregering og grænsen for kardinalitet.
- OpenTelemetry: Metrics data model — målere, summer, histogrammer, temporalitet og flaget for manglende værdi.
- Prometheus: Metric types — counter, gauge, histogram og summary.
- Prometheus: Metric and label naming — grundenheder, enhedssuffikser og `_total`.
- Prometheus: Histograms and summaries — hvor kvantiler beregnes, og hvilke der kan kombineres.
- Prometheus: Exposition formats — tekstformatet linje for linje.

Ordliste

Vælg ud fra jeres faktiske vilkår

Medbring udrulningsforløbet, driftsrisiciene og dem, der har vagten. Vi kan afgrænse en ændring med kontroller, en vej tilbage og tydeligt ansvar.

Se, hvad det koster Kontakt os →

Gør begreberne praktiske

Spørg, hvordan »Hvad er metrikker?« gælder i jeres organisation. Et konkret eksempel fra jeres nuværende system er nok til at begynde samtalen.

Se vores ydelser

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

Tag denne side med dig

Visitkort

Lavet med kærlighed af D.

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

Copyright © 2025 – nu · Engineer ApS · Alle rettigheder forbeholdes.

Udskrevet fra https://engineer.company/da/glossary/metrics/